LABORATORY FOR ENGINEERING MAN/MACHINE SYSTEMS
LEMS

System Identification and Model Reduction Using Modulating Function Techniques

*Yan Shen**

Technical Report LEMS-117
Division of Engineering, Brown University
March, 1993

*This work principally funded by NASA Grant NAG-1-1065.

Abstract of "System Identification and Model Reduction Using Modulating Function Techniques" by Yan Shen, Ph.D., Brown University, May 1993

Weighted least squares (WLS) and adaptive weighted least squares (AWLS) algorithms are initiated for continuous-time system identification using Fourier type modulating function techniques. Two stochastic signal models are examined using the mean square properties of the stochastic calculus: an equation error signal model with white noise residuals, and a more realistic white measurement noise signal model. The covariance matrices in each model are shown to be banded and sparse, and a joint likelihood cost function is developed which links the real and imaginary parts of the modulated quantities. The superior performance of above algorithms is demonstrated by comparing them with the LS/MFT and popular predicting error method (PEM) through 200 Monte Carlo simulations. A model reduction problem is formulated with the AWLS/MFT algorithm, and comparisons are made via six examples with a variety of model reduction techniques, including the well-known balanced realization method. Here the AWLS/MFT algorithm manifests higher accuracy in almost all cases, and exhibits its unique flexibility and versatility. Armed with this model reduction, the AWLS/MFT algorithm is extended into MIMO transfer function system identification problems. The impact due to the discrepancy in bandwidths and gains among subsystems is explored through five examples. Finally, as a comprehensive application, the stability derivatives of the longitudinal and lateral dynamics of an F-18 aircraft are identified using physical flight data provided by NASA. A pole-constrained SIMO and MIMO AWLS/MFT algorithm is devised and analyzed. Monte Carlo simulations illustrate its high-noise rejecting properties. Utilizing the flight data, comparisons among different MFT algorithms are tabulated and the AWLS is found to be strongly favored in almost all facets.

Acknowledgments

I would like to first extend my heartfelt gratitude to my advisor, Professor Allan E. Pearson. Without his meticulous and enlightening supervision, encouragement and patience, I would never have reached this point. I thank him sincerely not only for ushering me into such an entrancing research field: parameter identification, but also for his magnanimously granting me an invaluable opportunity to fulfill my Ph.D. dream at the most floundering moment of my life. I would also like to express my gratitude to the other members of my thesis committee, Professor Cooper and Professor McClure. I truly appreciate their thoughtful suggestions and comments.

I would like to specially thank Professor Sture K. F. Karlsson who provided the very first chance for me to be able to come to Brown. He enabled me to broaden the spectrum of my knowledge in many areas. The three years spent in his Lab together with Dr. Z. M. Yang and Dr. M. "TOUGHY" Rajaee were so wonderful and memorable. Those afternoon "negotiation" sessions at 37 Manning Street with "TOUGHY" have been always exhilarating.

I would also like to thank my colleagues in LEMS, especially A. Anthony Pandiscio, Alan Fullerton and Dr. J. Pan for those intriguing discussions, many of which have been more or less reflected at many places of this thesis.

Thanks must also go to Professor Andrea Simmons and Professor James Simmons of the Department of Psychology at Brown. The part-time "C" programming job they offered alleviated a severe financial pinch due to the sudden loss of a research

assistantship at one point. It was in Simmons' Lab that my dreariness had been completely diluted by the kindness and friendship from Dr. Tim Haresign, Mike Ferragamo, Dr. Steven Dear, Jan Wotton, Dr. Susan Allan, WBRU's "D.J." Johnny Fritz, Prestor Saillant and many others. Thank god, I was at the right place at the right moment with the right people !

Finally, I would like to express my gratitude to Professor Shu Wei at Tianjin University, P.R.China. He has reshaped my determination and enthusiasm towards science with his reverent devotion and dedication to research. I would also like to thank my parents for relentlessly instilling in me, since I was born, with Confucian dogmas such as "Among a million careers, being a scholar is the most prestigious". Their great expectation has been propelling me to overcome any possible barriers towards this goal. The generous help from my mother-in-law has contributed significantly to the completion of this work. My daughters, Ivy and Evy Shen, have been constantly intoxicating me through their "big hugs". Most of all, I would like to thank my wife, Qing Wang, who has co-authored every word, every sentence and every paragraph of this thesis with her great love, sacrifice and understanding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Mathematical models of dynamical systems are often required in engineering, physics, medicine, economics, ecology and in most areas of scientific enquiry. In control and system engineering, model-building or system identification from measurements of input-output data on a dynamical system has been one of the most active fields drawing enormous attention from researchers around the world. Among the many well established parameter estimation schemes, algorithms like the prediction error method (PEM) [1] [22] [12] have enjoyed a sustained boom in the past decade for discrete-time models. Although many researchers have been utilizing different kinds of transformations in an effort to link both continuous and discrete time model identification into a single framework, grim problems persist, like nonuniqueness of the

transformations, making these methods unwieldy and potentially unreliable. Hence, a direct attack on the problem is clearly preferred if a continuous time differential model is desired. The research presented in this thesis will be focused on the development of methodology in parameter identification for linear continuous-time differential equation models.

Generally, the identification of linear differential systems can be formulated in a deterministic vein using the classical steady state frequency domain approach for estimating the system transfer functions, or using a variety of time domain methods like Bellman-Kalaba's quasilinearization [3] [16], state variable filters, model reference techniques and adaptive observers [56]. In a stochastic vein, the known methods would include generalized least squares, instrumental variables, maximum likelihood and extended Kalman filtering techniques [56] [1].

Stemming from Shinbrot's method [51] [28] of moment functionals and using a set of carefully chosen modulating functions to facilitate converting a differential equation on a finite time interval into a set of algebraic regression equations in parameters, the *modulating function technique* (MFT) has been exploited as a tool for identifying continuous-time models. The modulating process itself can be viewed as discretizing a continuous-time differential system into a corresponding frequency domain characterization by means of a "resolving frequency" ω_0. Pearson and Lee [37] [18] [38] [35] utilized a set of real commensurable sinusoids $\{\cos m\omega_0 t, \sin m\omega_0 t\}$, $m = 0, 1, 2, \cdots, M$, where $\omega_0 = 2\pi/T$ is the resolving frequency or the "step size" in

the frequency domain, to build up the modulating function set through solving Vandermonde type linear equations[1] with which differentiating the data and estimating unknown initial conditions for time limited data can be totally avoided. By contrast, the other forms of modulating function methods, e.g., Hermite polynomials used by Takaya [55], Poisson process by Fairman and Shen [7] and Saha and Rao [43] [44] [45] [46], require either a long time interval of data or constrained initial conditions. Computationally, the modulating process by Fourier type modulating functions can be efficiently implemented by well documented Fast Fourier Transformation (FFT) algorithms while the other algorithms have to face a heavier numerical burden associated with the process of converting a differential equation on a finite time interval into a set of algebraic equations.

Besides the earlier work mentioned above, Co and Ydstie [6] have applied the trigonometric based Fourier type modulating function technique (FTMFT) to model reduction and some MIMO system identification problems[2] in chemical engineering. Pearson and Pan [36] [26] further expanded the FTMFT into the nonparametric identification framework, under which three least squares nonparametric algorithms for estimating system transfer functions are formed. Shen and Pearson [49] applied

[1]For numerical reasons, singular value decomposition (SVD) is strongly recommended [49] for the higher order modulating function sets. Pearson has suggested a much more efficient complex form of modulating function set (the set used in this thesis) which totally avoids tackling these Vandermonde linear equations.

[2]The identified MIMO models presented in that paper have higher orders than the actual MIMO systems, i.e., some unobservable modes were included in the models thus obtained.

the trigonometric-based MFT to analog Butterworth filter banks used in wind tunnel experiments in fluid dynamics, formulated Kalman filter type recursive schemes for both parametric and nonparametric LS/FTMFT algorithms, implemented the numerically sound Bierman's U-D matrix factorization algorithm for updating the high dimension Kalman gain matrix, developed the Parzen-window-based order determination algorithm for parametric LS/FTMFT, tested a parallel-adaptive memory-saving paradigm for the nonparametric LS/FTMFT method used in the on-line configurations, and thoroughly demonstrated the affect of higher order assumptions for both model and the modulating function set on the quality of the final estimation. Using the nonparametric approach, Pan [26] developed frequency matching model reduction algorithms which performed better than the parametric LS/FTMFT used by Co and Ydstie [6]. A FTMFT-based high resolution frequency estimation method [27] for signal processing has been proposed. Meanwhile the FTMFT has also been applied to the time-varying systems and nonlinear system identification problems [40] [29] [38] [31] [30] [32] [34] [33]. For brevity, FTMFT will be abbreviated as MFT in the rest of this thesis.

1.2 Organization of This Thesis

Following a brief introduction to MFT early in Chapter 2, one fundamental modulating property will be established showing that a modulated time domain white Gaussian stationary process will be a stationary Gaussian stochastic sequence in the

discrete frequency domain with its covariance matrix being banded by the order of the modulating function set. Then an idealistic equation error signal model is introduced leading to the first weighted least squares (WLS/MFT) algorithm which is based on a maximum likelihood estimate. For the much more realistic stochastic signal model with additive measurement noise, the explicit form of the regression error covariance matrix, which is a function of the unknown parameters, will likewise be shown to be banded but no longer stationary in the discrete frequency domain. Using this covariance matrix as a weighting, a numerical relaxation scheme dubbed as the adaptive weighted least squares (AWLS/MFT) algorithm will be devised, which is an approximated maximum likelihood estimate[3]. The third part of Chapter 2 deals with important implementation issues incurred by the use of a complex modulating function set. In order to combine both the real and imaginary parts into a unified cost function, one Lemma regarding vital relationships among the different covariance matrices is established as a prerequisite to constructing a joint likelihood cost function linking the information from the modulated real and imaginary quantities. In order to assuage the affliction caused by inverting the covariance matrix, a recur-

[3]The maximum likelihood estimate here is different from the ML estimate in [39] where multi-nonoverlaping data blocks are required, but in the AWLS/MFT, only one single data block is needed and this is perhaps a more economical and realistic framework especially in time limited transient I/O data. Meanwhile the framework of the AWLS/MFT can also guarantee a finer resolving frequency for the same short length of data. As a matter of fact, the AWLS/MFT algorithm was partly inspired and initiated so as to ease the urgency of efficiently utilizing rather limited available flight data in an aircraft identification problem (see Chapter 5).

sive banded-sparse matrix inversion algorithm will be derived and its computational efficiency and stability will be elucidated. Finally, a 200 Monte Carlo simulation comparison study is presented for the AWLS/MFT, WLS/MFT and LS/MFT, and a comparison is made with the popular prediction error method (PEM), where the improvement and superiority of WLS/MFT and AWLS/MFT algorithms will be illuminated. This chapter is the most important part of the thesis and will serve as the theoretical cornerstone for the rest of the thesis.

Armed with AWLS/MFT, Chapter 3 deals with model reduction. The performance of AWLS/MFT will be evaluated for six examples published in the literature providing comparisons with other known methods like nonparametric frequency fitting [26], FF-Padé approximation [14], time-domain optimization [47] and the well known Balanced-Realization (B-R) technique [24]. As one of the two best model reduction schemes in our comparison studies, AWLS/MFT is found to be able to perform at least as well as the B-R scheme, and in addition, possesses a kind of flexibility and versatility that the B-R algorithm lacks.

As the second application of the AWLS/MFT algorithm, the most general setting of MIMO system identification will be considered in Chapter 4. Inasmuch as the original transfer function matrix might not be modulatable directly in this case, one procedure is suggested for converting from an unmodulatable form to a higher order modulatable differential system. Based on the measurement noise signal model, the decomposability from MIMO into a set of MISO models is discussed from the view-

point of a joint likelihood cost function. AWLS/MFT is applied to each member of the set of MISO systems to get an identified model for the higher order system[4], and then the model reduction scheme of Chapter 3 is employed to obtain the original unmodulatable transfer function matrix. Five MISO systems with different bandwidth and magnitude combinations are used as examples to illustrate the impacts of these combinations on the accuracy of estimations and the overall feasibility and applicability of this approach. Results from 200 Monte Carlo runs under moderate additive noise settings have been quite encouraging.

In Chapter 5, flight data of an F-18 aircraft provided by NASA will be utilized, as a comprehensive application example of the WLS/MFT and AWLS/MFT algorithm, to identify the longitudinal and lateral dynamical systems of the aircraft. Due to the physical constraints posed by the aircraft itself, AWLS/MFT is first extended into a pole-constrained form, contrary to the decoupled form used in Chapter 4, and then employed to tackle the physical flight data. Simulation studies on these algorithms also manifest good noise-rejection features. Other extended AWLS/MFT algorithms based on the coupled state space models are devised as well, though they do not produce as good results as the I/O-based constrained AWLS/MFT algorithms.

[4]Co and Ydstie [6] used LS/MFT to accomplish this step and took it as the final result which actually leads to higher order models containing unobservable modes.

Chapter 2

Weighted Least Squares in MFT

2.1 Brief Introduction to the Basics of MFT

Consider the following nth order SISO differential equation system

$$\sum_{i=0}^{n} a_{n-i} y^{(i)}(t) = \sum_{i=0}^{n-1} b_{n-i} u^{(i)}(t) + e(t), \qquad a_0 = 1 \tag{2.1}$$

where $\{a_i\}$ and $\{b_i\}$,$i = 1, 2, ..., n$, are the time-invariant parameters needed to be identified from the input-output data pair $\{u(t), y(t); t \in [0, T]\}$ and superscript "(i)" means ith derivative, i.e., $y^{(0)}(t) = y(t)$ and $y^{(i)}(t) = d^i y(t)/dt^i$; $e(t)$ represents the effect of modeling errors. Assuming smoothness, the property which an n^{th} order modulating function $\phi(t)$ has to satisfy relative to a fixed time interval $[0, T]$ is:

$$\phi^{(i)}(0) = \phi^{(i)}(T) = 0; \qquad i = 0, 1, 2, ..., n-1. \tag{2.2}$$

Multiplying both sides of (2.1) with $\phi(t)$ and then integrating by-parts over $[0,T]$, while noting (2.2), leads to the following essential relation of the MFT:

$$\sum_{i=0}^{n}(-1)^i a_{n-i}\int_0^T y(t)\phi^{(i)}(t)dt = \sum_{i=0}^{n-1}(-1)^i b_{n-i}\int_0^T u(t)\phi^{(i)}(t)dt + \int_0^T e(t)\phi(t)dt, \quad a_0 = 1. \tag{2.3}$$

The consequences of (2.3) are: (i) modulating (2.1) with $\phi(t)$ has transferred the derivatives of the original data pair $\{u(t), y(t); t \in [0,T]\}$ into derivatives of a chosen **known** function $\phi(t)$, and (ii) the estimation of unknown initial conditions can be totally avoided due to (2.2).

Consider specifically the nth order complex Fourier type of modulating function set[1]:

$$\phi_{m,n}(t) = \frac{1}{T}e^{-jm\omega_0 t}(e^{-j\omega_0 t}-1)^n \tag{2.4}$$

$$= \frac{1}{T}\sum_{k=m}^{n+m} c_{k-m}e^{-jk\omega_0 t} \tag{2.5}$$

$$= \frac{1}{T}\sum_{k=0}^{n} c_k e^{-j(k+m)\omega_0 t} \tag{2.6}$$

$$m = 0, 1, 2, ..., M$$

$$c_k = (-1)^{n-k}\begin{pmatrix} n \\ k \end{pmatrix} \tag{2.7}$$

where ω_0 is called the resolving frequency defined as $\omega_0 = 2\pi/T$ and T is the time interval of the data block.

Applying the above modulating function set to (2.3) leads to the following regression model

$$\gamma_0^y(m) = \gamma(m)\theta + \epsilon_n(m), \qquad m = 0, 1, 2, ..., M \tag{2.8}$$

[1]The equivalence between (2.4) and (2.5)~ (2.7) follows from the binomial expansion.

with the regressor row vector: $\gamma(m) = [\gamma_1^y(m), ..., \gamma_n^y(m), \gamma_1^u(m), ..., \gamma_n^u(m)]$ in which

$$\gamma_i^f(m) = (-1)^{n-i} \int_0^T f(t)\phi_{m,n}^{(n-i)}(t)dt \tag{2.9}$$

$$i = 0, 1, 2, ..., n; \qquad f(t) = \{\ y(t) \quad \text{or} \quad u(t).\ \}$$

The parameter vector θ and the model error $\epsilon_n(m)$ are defined respectively as

$$\theta = \begin{pmatrix} -a_1 \\ \vdots \\ -a_n \\ b_1 \\ \vdots \\ b_n \end{pmatrix} \tag{2.10}$$

$$\epsilon_n(m) = \int_0^T e(t)\phi_{m,n}(t)dt. \tag{2.11}$$

Introducing the following notation:

$$Y = (\gamma_0^y(0), \gamma_0^y(1), ..., \gamma_0^y(M))^T \tag{2.12}$$

$$\Gamma = \begin{pmatrix} \gamma(0) \\ \gamma(1) \\ \vdots \\ \gamma(M) \end{pmatrix} \tag{2.13}$$

$$\epsilon = (\epsilon_n(0), \epsilon_n(1), ..., \epsilon_n(M))^T \tag{2.14}$$

the relation (2.8) can be rewritten into a vector and complex-valued regression form

$$Y = \Gamma\theta + \epsilon \tag{2.15}$$

2.2 MFT Under a Stochastic Framework

2.2.1 Mathematical Antecedent and Modulated White Noise

Stochastic Calculus in Mean Square Sense

Due to the involvement of stochastic processes, all the stochastic calculus operators including limits, continuity, integration and differentiation in this dissertation are presumed to be carried out in the **mean square** (m.s.) sense (refer to Appendix D for details). In light of Appendix D, especially Corollary 1 and Corollary 2, all the above established modulating properties and relations hold true as long as $\{u(t), y(t), e(t)\}$ are n^{th} order m.s. differentiable.

Maximum Likelihood Estimate

Armed with this prerequisite, some further elaborations on (2.15) can be continued. In order to handle the complex regression form more efficiently, we shall examine the real regression form first. Later on, in the implementation section of this Chapter, the process of converting a complex regression form into a real regression form will be scrutinized. For a **real** regression equation in the form of (2.15), provided the equation error ϵ has jointly Gaussian $\mathcal{N}(0, W)$ distribution, the log-likelihood function of ϵ can be written as [11]

$$\mathcal{L}(\theta, W) = -\frac{M}{2}\ln(2\pi) - \frac{\ln|W|}{2} - \frac{1}{2}(Y - \Gamma\theta)^T W^{-1}(Y - \Gamma\theta) \qquad (2.16)$$

If the covariance matrix W is **known** , then from the necessary condition for maximizing the likelihood function, $\frac{\partial \mathcal{L}}{\partial \theta} = 0$, the well known weighted least squares estimate of θ can be written as

$$\hat{\theta} = (\Gamma^T W^{-1} \Gamma)^{-1} \Gamma^T W^{-1} Y \tag{2.17}$$

and this estimate will be the maximum likelihood estimate of θ or the minimax entropy estimate [11].

Another nice property, which might be used in the system order determination problem, can be stated in the following Lemma:

Lemma 1 (distribution of posteriori cost function) *Let the cost function $J(\theta)$ be defined as*

$$J(\theta) = (Y - \Gamma^T \theta)^T W^{-1} (Y - \Gamma^T \theta)$$

then its corresponding WLS estimate is

$$\hat{\theta} = (\Gamma^T W^{-1} \Gamma)^{-1} \Gamma^T W^{-1} Y. \tag{2.18}$$

If the covariance matrix W of the sequence $\{\epsilon_n(m)\}$, $m = 0, 1, 2, ..., M$, is known, then for the posteriori cost function $J(\hat{\theta})$,

$$J(\hat{\theta}) \sim \chi^2(M + 1 - \tilde{n}) \tag{2.19}$$

where symbol $\sim$ means "obeys" and $\tilde{n}$ is the dimension of the parameter vector θ.

Proof: Define output error residuals $\hat{\epsilon}$ as

$$\hat{\epsilon} = Y - \Gamma \hat{\theta}. \tag{2.20}$$

Putting (2.18) into the above definition:

$$\hat{\epsilon} = [I - \Gamma(\Gamma^T W^{-1}\Gamma)^{-1}\Gamma^T W^{-1}]\epsilon \tag{2.21}$$

Decompose $W^{-1} = W^{-\frac{1}{2}} \cdot W^{-\frac{1}{2}}$, and apply linear filter $W^{-\frac{1}{2}}$ to the residual $\hat{\epsilon}$:

$$\begin{aligned} W^{-\frac{1}{2}}\hat{\epsilon} &= W^{-\frac{1}{2}}[I - \Gamma(\Gamma^T W^{-1}\Gamma)^{-1}\Gamma^T W^{-1}]\epsilon \\ &= \underbrace{[I - W^{-\frac{1}{2}}\Gamma(\Gamma^T W^{-1}\Gamma)^{-1}\Gamma^T W^{-\frac{1}{2}}]}_{P} \cdot \underbrace{W^{-\frac{1}{2}}\epsilon}_{X} \end{aligned} \tag{2.22}$$

Directly from the above definition of P, we can prove that P is an idempotent matrix

$$P^2 = P.$$

From Lemma A.28 [54] we know that the following statements hold for an arbitrary idempotent matrix P:

- All eigenvalues are either zero or one
- Rank(P)=tr$P$

Then in our case

$$\begin{aligned} Rank(P) &= \mathrm{tr}P \\ &= \mathrm{tr}[I - W^{-\frac{1}{2}}\Gamma(\Gamma^T W^{-1}\Gamma)^{-1}\Gamma^T W^{-\frac{1}{2}}] \\ &= M + 1 - \mathrm{tr}[(\Gamma^T W^{-1}\Gamma)^{-1}\Gamma^T W^{-\frac{1}{2}} W^{-\frac{1}{2}}\Gamma] \\ &= M + 1 - \tilde{n} \end{aligned}$$

where $\tilde{n}$ is the dimension of θ.

Paraphrasing from Lemma B.8 [54]:

Assume $X \sim \mathcal{N}(m, W)$ and set $Z = AX + B$ for constant A and B of appropriate dimensions. Then $Y \sim \mathcal{N}(Am + b, AWA^T)$.

Therefore, for the filtered sequence $X = W^{-\frac{1}{2}}\epsilon$,

$$X = W^{-\frac{1}{2}}\epsilon \sim \mathcal{N}(0, I) \tag{2.23}$$

$$J(\hat{\theta}) = \hat{\epsilon}^T W^{-1} \hat{\epsilon} = X^T P X. \tag{2.24}$$

Paraphrasing from Lemma B.13 [54]:

Let X be an M-dimensional Gaussian vector, $X \sim \mathcal{N}(0, I)$ and let P be an $(M|M)$-dimensional idempotent matrix of rank $M - \tilde{n}$. Then $X^T P X$ is $\chi^2(M - \tilde{n})$ distributed.

Directly applying the above Lemma to $J(\hat{\theta})$, we have $J(\hat{\theta}) \sim \chi^2(M + 1 - \tilde{n})$. The proof is complete.

In most identification problems, the covariance matrix W is not available in advance. But if some knowledge or assumptions about the statistics of the regression model error ϵ can be imposed beforehand, it is indeed possible, as shown in the future sections, to derive an explicit form of W which may or may not depend on the parameters θ.

Stationary White Gaussian Noise

Time domain stationary white Gaussian noise is a symbolic process $n(t) = \dot{W}_0(t)$ (where $W_0(t)$ is a Wiener-Levy process) with mean function

$$E\{n(t)\} = 0$$

and correlation function

$$R_n(t_1, t_2) = \delta(t_1 - t_2)$$

where δ is the Dirac delta function. From the definition of m.s. derivative listed in Appendix D, this white Gaussian noise is not m.s. differentiable at all. However, after some dedicated and lengthy mathematical maneuvers, e.g., pp 313~328 in [17], a certain justification for its wide usage can be made.

.....

> To conclude our discussion, let us summarize what we have gained by introducing the concept of white Gaussian noise. For one thing, it allows us to apply the rules of MS calculus even to processes that are not MS differentiable, and it greatly simplifies calculations involving the Wiener integral. $\cdots$. More importantly, however, the white Gaussian noise represents an idealized form of a *continuous physical noise* just like the Dirac delta function is an idealized form of a unit impulse. Thus, whenever we wish to model a physical noise that in reality may consist of densely packed narrow impulses of constant energy and random polarity, we may reach for a white Gaussian noise as a

suitable mathematically tractable idealization.

— H.J. Larson and B.O. Shubert (pp 327 in [17])

Another vivid example of approximating $W_0(t)$ on $[0,T]$ with a sequence of processes, $W^{(k)}(t); k = 1,2,...$, was illustrated in [42], pp 94~97, where it was proved: (i) the sample functions of $W^{(k)}(t)$ are infinitely often differentiable on $[0,T]$ with probability one, (ii) the sequence $W^{(k)}(t)$ is smooth in the m.s. sense on $[0,T]$, (iii) $E\{W^{(k)}(t)\} = 0$, $t \in [0,T]$, (iv) $W^{(k)}(t)$ is normally distributed, (v) If $t_1 \leq t_2 < t_3 \leq t_4$, the increments of $W^{(k)}(t)$ on $[t_1,t_2)$ and $[t_3,t_4)$ are orthogonal for sufficiently large k, (vi) As $k \to \infty$, $W^{(k)}(t) \to W(t)$ in m.s. uniformly in $t \in [0,T]$, (vii) $E\{W^{(k)}(s)W^{(k)}(t)\} \to E\{W(s)W(t)\}$ as $k \to \infty$. Hence $W^{(k)}(t)$ for a sufficiently large k will not only be m.s. infinitely differentiable, but also infinitely close to the ideal Wiener-Levy process. In the rest of this thesis, we therefore view that the white Gaussian noise is actually defined as $n(t) = \dot{W}^{(k)}(t)$, $k \to \infty$.

Modulated White Gaussian Noise

From the discussion in 2.1, we can say that modulating a time domain process using a set of modulating functions is equivalent to applying a linear transformation to it. For Gaussian distributed random processes, one well known fact is that any linear operation performed on a Gaussian process results in another Gaussian process [17]. In our case, the modulating affect on a white Gaussian noise $e(t)$, $t \in [0,T]$, can be exhibited in the following Lemma:

Lemma 2 (modulated white noise) *If $e(t)$ is white Gaussian noise in the continuous time domain with $E[e(t_1) \cdot e(t_2)] = \sigma^2 \cdot \delta_D(t_1 - t_2)$ and $\epsilon_n(m)$ is the result of modulating $e(t)$ by the n^{th} order modulating function set $\phi_{m,n}(t)$ defined in* (2.6)*, i.e.,*

$$\epsilon_n(m) = \int_0^T \phi_{m,n}(t) e(t) dt$$

then the sequence $\{\epsilon_n(m)\}$, $m = 0, 1, 2, ..., M$, is a stationary discrete Gaussian process; its covariance matrix $W_e = E(\epsilon^* \epsilon^T)$ is a banded Toeplitz matrix with bandwidth n, each element of which is real and can be expressed as

$$w_{e_{(m,m+l)}} = E[\epsilon_n(m)\epsilon_n^*(m+l)] = \begin{cases} 0 & |l| > n \\ \frac{\sigma^2}{T} \frac{(-1)^l \cdot (2n)!}{(n-l)!(n+l)!} & |l| \leq n \end{cases} \tag{2.25}$$

Proof: From the definition and Corollary 3 in Appendix D

$$\begin{aligned} E[\epsilon_n(m)\epsilon_n^*(m+l)] &= E[\int_0^T e(t_1) \cdot \phi_{m,n}(t_1) dt_1 \cdot \int_0^T e(t_2) \cdot \phi_{m+l,n}^*(t_2) dt_2] \\ &= \int_0^T \int_0^T E[e(t_1) \cdot e(t_2)] \cdot \phi_{m,n}(t_1) \cdot \phi_{m+l,n}^*(t_2) dt_1 dt_2. \end{aligned} \tag{2.26}$$

Noting that $E[e(t_1) \cdot e(t_2)] = \sigma^2 \cdot \delta_D(t_1 - t_2)$ and utilizing the sifting property:

$$\int_0^T g(t) \cdot \delta_D(t - \tau) dt = g(\tau)$$

for any continuous function $g(t)$, equation (2.26) can be written as

$$E[\epsilon_n(m)\epsilon_n^*(m+l)] = \sigma^2 \int_0^T \phi_{m,n}(t) \phi_{m+l,n}^*(t) dt. \tag{2.27}$$

Considering the relations (2.4), (2.5), and (2.6):

$$\begin{aligned} \phi_{m,n}(t)\phi_{m+l,n}^*(t) &= \frac{1}{T^2}[\sum_{k_1=0}^{n} c_{k_1} e^{-j(k_1+m)\omega_0 t}] \cdot [\sum_{k_2=0}^{n} c_{k_2} e^{j(k_2+m+l)\omega_0 t}] \\ &= \frac{1}{T^2} \sum_{k_1=0}^{n} \sum_{k_2=0}^{n} c_{k_1} c_{k_2} e^{j[l+(k_2-k_1)]\omega_0 t} \end{aligned} \tag{2.28}$$

and substituting this into (2.27)

$$E[\epsilon_n(m)\epsilon_n^*(m+l)] = \frac{\sigma^2}{T^2}\sum_{k_1=0}^{n}\sum_{k_2=0}^{n} c_{k_1}c_{k_2}\int_0^T e^{j[l+(k_2-k_1)]\omega_0 t}dt. \tag{2.29}$$

But only the frequency indices satisfying $l + k_2 - k_1 = 0$ will contribute to the above integral, i.e.,

$$E[\epsilon_n(m)\epsilon_n^*(m+l)] = \begin{cases} 0 & |l| > n \\ \frac{\sigma^2}{T}\sum_{k=0}^{n-l} c_k c_{k+l} & |l| \le n \end{cases} \tag{2.30}$$

and from formula $O.156.2$ in [13]

$$\begin{aligned} \sum_{k=0}^{n-l} c_k c_{k+l} &= \sum_{k=0}^{n-l}(-1)^l \cdot \binom{n}{k} \cdot \binom{n}{k+l} \\ &= \frac{(-1)^l \cdot (2n)!}{(n-l)!(n+l)!} \end{aligned} \tag{2.31}$$

Combining the above two equations, we have

$$w_{e_{(m,m+l)}} = E[\epsilon_n(m)\epsilon_n^*(m+l)] = \begin{cases} 0 & |l| > n \\ \frac{\sigma^2}{T}\frac{(-1)^l\cdot(2n)!}{(n-l)!(n+l)!} & |l| \le n \end{cases} \tag{2.32}$$

Therefore, the covariance matrix is banded with bandwidth n and each element is just a real function of l so that the sequence is at least a wide sense stationary (w.s.s) process. But for a w.s.s. Gaussian processes, it must be stationary. Equation (2.32) also has manifested the fact that the covariance matrix W_e is a Toeplitz matrix. The proof is complete.

More generally, for any jointly Gaussian distributed time domain stationary process, its modulated sequence would still be a stationary process, except that the bandwidth of the covariance matrix is not necessarily equal to the order (n) of the n^{th} order modulating function set.

2.2.2 WLS Algorithm and Equation Error Signal Model

Consider the idealized equation error signal model on $[0,T]$:

$$\sum_{i=0}^{n} a_{n-i} y^{(i)}(t) = \sum_{i=0}^{n-1} b_{n-i} u^{(i)}(t) + e(t), \qquad a_0 = 1 \tag{2.33}$$

where the equation error $e(t)$ is **white Gaussian noise**, and the $\{a_i\}$ and $\{b_i\}$, $i = 1, 2, ..., n$, are the time-invariant parameters needed to be identified from the input-output data pair $\{u(t), y(t); t \in [0,T]\}$. Using the same notation, the corresponding modulated equation error model can be put into the same regression form:

$$Y = \Gamma\theta + \epsilon \tag{2.34}$$

where the error vector $\epsilon = [\epsilon_n(0), ..., \epsilon_n(M)]^T$ results from modulating the white Gaussian noise $e(t)$ using the n^{th} order modulating function set $\{\phi_{m,n}(t), m = 0, 1, ..., M\}$. For the error sequence $\{\epsilon_n(m)\}$, based on Lemma 2, its covariance matrix will be $W_e = E(\epsilon^* \epsilon^T)$ which in this ideal setting is not related to the system parameters or input/output data and hence can be written out explicitly in the form of (2.32). With this covariance matrix as weighting and the discussion in 2.2.1, a maximum likelihood estimate can be obtained and framed into the following WLS/MFT algorithm[2]:

Algorithm 1 (WLS/MFT Estimate)

1. *Build the weighting matrix* $W_e = E(\epsilon^* \epsilon^T)$ *based on Lemma* 2.

[2]For a detailed implementation of this algorithm, please refer to Section 2.3

2. Compute the parameter vector from

$$\hat{\theta}_{WLS} = (\Gamma^T W_e^{-1}\Gamma)^{-1}\Gamma^T W_e^{-1} Y. \tag{2.35}$$

Remark: In most cases the variance σ^2 of the noise $e(t)$ is not known even though it has no affect on the above algorithm due to cancellation in (2.35). Thus, if we rewrite $W_e = \frac{\sigma^2}{T} \cdot \underline{W_e}$, where $\underline{W_e}$ is defined only by binomial coefficients and the order n, then (2.35) can be adjusted accordingly as

$$\hat{\theta}_{WLS} = (\Gamma^T \underline{W_e}^{-1}\Gamma)^{-1}\Gamma^T \underline{W_e}^{-1} Y. \tag{2.36}$$

2.2.3 Measurement Noise Signal Model and AWLS/MFT Algorithms

Measurement Noise Signal Model

Different from the equation error signal model, the measurement noise signal model shown in Figure 2.1 can be characterized as the ideal input/output data pair $\{\tilde{u}(t), \tilde{y}(t)\}$ contaminated with additive white noises $v(t)$ and $n(t)$. Our goal here is to identify the parameters of model $H(s)$ from this contaminated data pair $\{\tilde{u}(t)+v(t), \tilde{y}(t)+n(t)\}$. Assume the model $H(s)$ in the time domain is of the differential form

$$\sum_{i=0}^{n} a_{n-i}\tilde{y}^{(i)}(t) = \sum_{i=0}^{n-1} b_{n-i}\tilde{u}^{(i)}(t) \qquad a_0 = 1. \tag{2.37}$$

Then from Figure 2.1

$$\tilde{u}(t) = u(t) - v(t) \tag{2.38}$$

$$\tilde{y}(t) = y(t) - n(t). \tag{2.39}$$

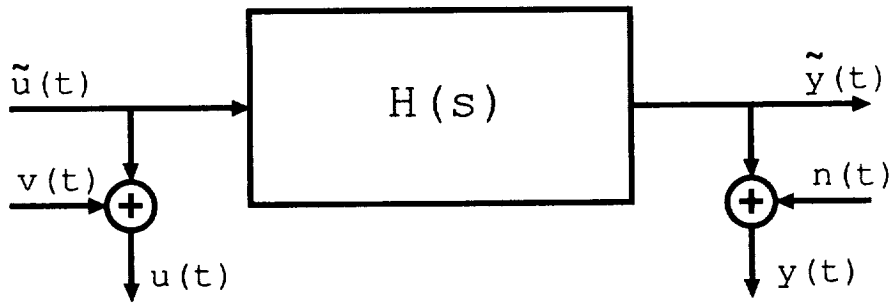

Figure 2.1: Measurement noise signal model.

Substituting $\tilde{u}(t)$ and $\tilde{y}(t)$ into model (2.37):

$$\sum_{i=0}^{n} a_{n-i} y^{(i)}(t) = \sum_{i=0}^{n-1} b_{n-i} u^{(i)}(t) + \underbrace{\sum_{i=0}^{n} a_{n-i} n^{(i)}(t) - \sum_{i=0}^{n-1} b_{n-i} v^{(i)}(t)}_{e(t)} \tag{2.40}$$

Unlike the equation error model, $e(t)$ is directly related to the parameters needed to be identified.

Covariance Matrix of Modulated Error Sequence

Before modulating the above equation, let us observe a general relation for any n^{th} order differentiable function $f(t)$:

$$\begin{aligned}
\int_0^T f^{(i)}(t) \cdot \phi_{m,n}(t) dt &= (-1)^i \cdot \int_0^T f(t) \cdot \phi_{m,n}^{(i)}(t) dt \\
&= \frac{1}{T}(-1)^i \cdot \int_0^T \sum_{k=m}^{n+m} (-jk\omega_0)^i c_{k-m} f(t) e^{-jk\omega_0 t} dt \\
&= \frac{1}{T} \sum_{k=m}^{n+m} (jk\omega_0)^i c_{k-m} \cdot \underbrace{\int_0^T f(t) e^{-jk\omega_0 t} dt}_{F(k)} \\
&= \frac{1}{T} \underbrace{\sum_{k=m}^{n+m} (jk\omega_0)^i c_{k-m} F(k)}_{F_i(m)}. \qquad (2.41) \\
&= \frac{1}{T} F_i(m) \qquad (2.42)
\end{aligned}$$

Armed with this relation, the modulated $e(t)$ becomes

$$\begin{aligned}\epsilon(m)&=\int_0^T e(t)\cdot\phi_{m,n}(t)dt\\&=\int_0^T\left[\sum_{i=0}^{n}a_{n-i}n^{(i)}(t)-\sum_{i=0}^{n-1}b_{n-i}v^{(i)}(t)\right]\cdot\phi_{m,n}(t)dt\\&=-(N_n(m),N_{n-1}(m),...,N_0(m),V_{n-1}(m),...,V_0(m))\begin{pmatrix}-1\\ \theta\end{pmatrix}.\end{aligned}\tag{2.43}$$

Partition θ in (2.10) as

$$\theta=\begin{pmatrix}\theta_a\\ \theta_b\end{pmatrix}.\tag{2.44}$$

Then from equation (2.43) and (2.41)

$$\begin{aligned}\epsilon(m)&=(N_n(m),N_{n-1}(m),...,N_0)\begin{pmatrix}1\\ -\theta_a\end{pmatrix}+(V_{n-1}(m),...,V_0(m))(-\theta_b)\\&=\frac{1}{T}\sum_{k=m}^{n+m}\underbrace{c_{k-m}\left((jk\omega_0)^n,...,(jk\omega_0)^0\right)\begin{pmatrix}1\\ -\theta_a\end{pmatrix}}_{\alpha(k,m,\theta_a)}\int_0^T n(t)e^{-jk\omega_0 t}dt\\&\quad+\frac{1}{T}\sum_{k=m}^{n+m}\underbrace{c_{k-m}\left((jk\omega_0)^{n-1},...,(jk\omega_0)^0\right)(-\theta_b)}_{\beta(k,m,\theta_b)}\int_0^T v(t)e^{-jk\omega_0 t}dt\\&=\frac{1}{T}\sum_{k=m}^{n+m}\alpha(k,m,\theta_a)\int_0^T n(t)e^{-jk\omega_0 t}dt+\frac{1}{T}\sum_{k=m}^{n+m}\beta(k,m,\theta_b)\int_0^T v(t)e^{-jk\omega_0 t}dt\end{aligned}\tag{2.45}$$

which serves to define the parameter dependent frequency functions $\alpha(k,m,\theta_a)$ and $\beta(k,m,\theta_b)$. If the statistics of $n(t)$ and $v(t)$ are known, the above relation would be the starting point of computing the covariance matrix of the residual error frequency sequence $\{\epsilon(m),m=0,1,2,...,M\}$. In a special case where $n(t)$ and $v(t)$ are mutually independent and (approximately) white Gaussian, we have the following Lemma:

Lemma 3 *If $n(t)$ and $v(t)$ are mutually independent white Gaussian noises, then the covariance matrix W_s of the modulated error sequence $\{\epsilon(m),\ m = 0,1,2,...,M\}$ is banded with the order of the modulating functions as bandwidth, and its $(m, m+l)$ element denoted as $w_{s_{(m,m+l)}}$ is real and can be expressed as*

$$\begin{aligned} w_{s_{(m,m+l)}} &= E[\epsilon(m)\epsilon^*(m+l)] \\ &= \begin{cases} 0 & |l| > n \\ \frac{\sigma_n^2}{T}\sum_{k_1=0}^{n-l}\alpha(k_1+m+l,m,\theta_a)\alpha^*(k_1+m+l,m+l,\theta_a) & \\ \quad +\frac{\sigma_v^2}{T}\sum_{k_2=0}^{n-l}\beta(k_2+m+l,m,\theta_b)\beta^*(k_2+m+l,m+l,\theta_b) & |l| \le n \end{cases} \end{aligned} \tag{2.46}$$

or in brevity

$$W_s = g(\theta) \qquad \textit{and} \qquad W_s \textit{ is real.} \tag{2.47}$$

Proof: Denoting the first term of $E[\epsilon(m)\epsilon^*(m+l)]$ by W_{1st}:

$$W_{1st}=\frac{1}{T^2}\sum_{k_3=m}^{n+m}\sum_{k_4=m+l}^{n+m+l}\alpha(k_3,m,\theta_a)\cdot\alpha^*(k_4,m+l,\theta_a)\int_0^T\int_0^T E[n(t_1)\cdot n(t_2)]e^{-jk_3\omega_0 t_1+jk_4\omega_0 t_2}dt_1dt_2$$

For the white noises $E[n(t_1)\cdot n(t_2)] = \sigma_n^2\cdot\delta_D(t_1-t_2)]$, this leads to

$$W_{1st} = \frac{1}{T^2}\sum_{k_3=m}^{n+m}\sum_{k_4=m+l}^{n+m+l}\alpha(k_3,m,\theta_a)\alpha^*(k_4,m+l,\theta_a)\int_0^T e^{-j(k_3-k_4)\omega_0 t}dt.$$

Letting $k_5 = k_3 - m$, and $k_6 = k_4 - m - l$, we have

$$W_{1st}=\frac{1}{T^2}\sum_{k_5=0}^{n}\sum_{k_6=0}^{n}\alpha(k_5+m,m,\theta_a)\alpha^*(k_6+m+l,m+l,\theta_a)\int_0^T e^{-j(k_5-k_6-l)\omega_0 t}dt.$$

Replacing k_6 with k_1 and noting that the integral is nonzero only for $k_5 - k_6 - l = 0$,

$$W_{1st} = \begin{cases} 0 & |l| > n \\ \frac{\sigma_n^2}{T}\sum_{k_1=0}^{n-l}\alpha(k_1+m+l,m,\theta_a)\alpha^*(k_1+m+l,m+l,\theta_a) & |l| \le n \end{cases}$$

From the independence assumption, the second term of $E[\epsilon(m)\epsilon^*(m+l)]$ can be similarly shown to be

$$W_{2nd} = \begin{cases} 0 & |l| > n \\ \frac{\sigma_w^2}{T} \sum_{k_2=0}^{n-l} \beta(k_2+m+l, m, \theta_b)\beta^*(k_2+m+l, m+l, \theta_b) & |l| \leq n \end{cases}$$

Based on the definitions of $\alpha(k, m, \theta_a)$ and $\beta(k, m, \theta_b)$ in (2.45), the simple yet important facts that

$$\begin{cases} \alpha(k, m, \theta_a) \cdot \alpha^*(k, \bar{m}, \theta_a) & \geq & 0 \\ \beta(k, m, \theta_b) \cdot \beta^*(k, \bar{m}, \theta_b) & \geq & 0 \end{cases} \qquad \text{for all } m \text{ and } \bar{m} \qquad (2.48)$$

can be easily drawn. Thus, combining W_{1st} and W_{2nd} with (2.48) yields (2.46) and (2.47). This proves Lemma 3.

AWLS Algorithms without/with Stability Constraint

Due to the dependence of $\alpha(\cdot)$ and $\beta(\cdot)$ in the covariance matrix on the parameters desired to be identified, the maximum likelihood estimate discussed in Section 2.2.2 cannot be implemented directly. From a numerical point of view, we have the following posed problem:

> How to find solutions for $\hat{\theta}_{AWLS}$ and $\hat{W}_s$ from the implicit nonlinear equation set

$$\hat{\theta}_{AWLS} = (\Gamma^T \hat{W}_s^{-1} \Gamma)^{-1} \Gamma^T \hat{W}_s^{-1} Y \qquad (2.49)$$

$$\hat{W}_s = g(\hat{\theta}_{AWLS}). \qquad (2.50)$$

To solve it, the following relaxation scheme, which will be referred to here as the adaptive weighted least-squares (AWLS) algorithm, can be constructed; it can be

viewed as an approximate maximum likelihood estimate and is similar to the algorithm presented in pp 47∼50 of [11] in terms of numerical relaxation:

Algorithm 2 (AWLS/MFT estimate) [3]

1. *Choose an error tolerance* μ *for convergence judgement and initialize the step index* $i = 0$.

2. *Estimate initial* $\hat{\theta}^0_{AWLS}$ *with identity matrix* I *or* W_e *in* (2.49).

3. *Give an initial estimate of* $\hat{W}^0_s$ *from* (2.50) *where* $g(\theta)$ *is defined by* (2.45)∼(2.47).

4. *i=i+1.*

5. *Estimate* $\hat{\theta}^i_{AWLS}$ *from* (2.49) *with* $\hat{W}^{i-1}_s$.

6. *Compute* $\hat{W}^i_s$ *from* (2.50) *with* $\hat{\theta}^i_{AWLS}$.

7. *If* $\|\hat{\theta}^i_{AWLS} - \hat{\theta}^{i-1}_{AWLS}\| \leq \mu$, *stop; otherwise continue.*

8. *Go back to step* 4.

Remarks on the above AWLS/MFT Algorithm:

1. Unlike the residuals in Lemma 2, the sequence $\{\epsilon(m)\}$, $m = 0, 1, ..., M$, for the residuals in Lemma 3, is no longer stationary.

2. Because only a biased estimate $\hat{\theta}_{AWLS}$ can be obtained, so likewise the covariance matrix will be biased. Therefore, the AWLS estimate is no longer the

[3] For a detailed implementation of this algorithm, please refer to Section 2.3

exact maximum likelihood estimate. But when the bias is small, it is still close to the maximum likelihood estimate.

3. The linear filter $\hat{W}_s^{\frac{1}{2}}$ is no longer a whitening filter as in Lemma 1. But when the bias is small, Lemma 1 still holds approximately.

4. Since implementing the algorithm depends only on the *ratio* of the pair (σ_n^2, σ_v^2), it can still be implemented for unknown noise levels when either σ_n^2 or σ_v^2 is zero, i.e., one is negligible relative to the other, or when $\sigma_n^2 \approx \sigma_v^2$.

5. When the distributions of $n(t)$ and $v(t)$ are not known beforehand, the above algorithm still could be applied and in almost all our cases gives better results than the LS/MFT and WLS/MFT algorithms.

6. Numerical experiments show that if the algorithm converges, it will converge to the same value no matter from which initial weighting, i.e., either from LS or from WLS, even though the rate of convergence could be different[4].

7. The AWLS algorithm is much less sensitive to the chosen modulating bandwidth ω_B, implying that it is a more robust algorithm[5].

---

[4]Using the estimate of θ with the WLS/MFT algorithm to estimate $\hat{W}_s^0$ is more likely to lead to a faster convergence.

[5]This will be further illustrated in 5.4.2. Please refer to Pearson and Lee [39] for guidelines in determining the modulating bandwidth ω_B and the resolving frequency ω_0.

8. As for convergence of the above algorithm, the comment from Goodwin and Payne (see pp 50 of [11]) regarding their algorithm seems also valid for this algorithm: "Unfortunately the authors are not aware of the existence of a global convergence proof for the above relaxation algorithm. However, computational studies indicate that the algorithm works well in practice." As a matter of fact, Algorithm 2 has never failed to converge in all our numerical study examples.

In most control systems, the estimated model is required to be stable. Inspired by the Projection Algorithm on pp 367 of [22], the Algorithm 2 could be easily adjusted according to the following form so that the estimated model with stable poles can be essentially guaranteed.

Algorithm 3 (AWLS/MFT estimate with stability constraint)

1. *Choose an error tolerance μ for convergence judgement and initialize the step index $i = 0$.*

2. *Estimate the initial $\hat{\theta}^0_{AWLS}$ with identity matrix I or W_e in* (2.49).

3. *Obtain an initial estimate of $\hat{W}_s^0$ from* (2.50) *where $g(\theta)$ is defined by* (2.45)$\sim$(2.47).

4. *i=i+1.*

5. *Estimate $\hat{\theta}^i_{AWLS}$ from* (2.49) *with $\hat{W}_s^{i-1}$.*

6. *If $\hat{\theta}^i_{AWLS}$ is not stable,i.e., the polynomial $\sum_{i=0}^{n} \hat{a}_{n-i}s^i$ is not Hurwitzian, mirror the unstable poles into the left-half-plane and recalculate $\hat{\theta}^i_{AWLS}$; otherwise skip this step.*

7. *Compute* $\hat{W}_s^i$ *from* (2.50) *with* $\hat{\theta}^i_{AWLS}$.

8. *If* $\|\hat{\theta}^i_{AWLS} - \hat{\theta}^{i-1}_{AWLS}\| \leq \mu$ *stop; otherwise continue.*

9. *Go back to step* 4.

Remark: In most cases, this adjustment is not necessary. However, when there exist marginally unstable poles, the above algorithm has been proven to be very effective (as a typical example, see Section 5.4.2).

2.3 Implementing WLS/MFT and AWLS/MFT

As mentioned earlier, both the real and imaginary parts of Definition (2.4) are still modulating functions in themselves. Therefore, modulating a differential system will provide two sets of algebraic equations corresponding to the modulated real and imaginary parts respectively. In this section, we shall devise a joint cost function which can utilize both parts and can be easily minimized under the weighted least squares framework. But first, some basic relations bridging the real and imaginary parts should be explored and disclosed.

2.3.1 Basic Relation of Covariance Matrices

Considering the complex modulating function set (2.4) defined as

$$\phi_{m,n}(t) = \frac{1}{T} e^{-jm\omega_0 t}(e^{-j\omega_0 t} - 1)^n \tag{2.51}$$

where n is the order of modulating function set and $m = 0, 1, 2, ..., M$, it can be separated into a real part $\phi^R_{m,n}(t)$ and imaginary part $\phi^I_{m,n}(t)$:

$$\phi_{m,n}(t) = \phi^R_{m,n}(t) + j\phi^I_{m,n}(t). \tag{2.52}$$

Denoting the signal quantities modulated by $\phi^R_{m,n}(t)$ with subscript R and those modulated by $\phi^I_{m,n}(t)$ with subscript I, we have (cf. (2.12)~(2.15))

$$Y = Y_R + jY_I \tag{2.53}$$

$$\Gamma = \Gamma_R + j\Gamma_I \tag{2.54}$$

$$\epsilon = \epsilon_R + j\epsilon_I. \tag{2.55}$$

The regression $Y = \Gamma\theta + \epsilon$ can also be divided into two parts such that

$$\epsilon_R = Y_R - \Gamma_R\theta \tag{2.56}$$

$$\epsilon_I = Y_I - \Gamma_I\theta \tag{2.57}$$

Further, define four different covariance matrices (W, W_R, W_I and W_{RI})[6] with their $(m, m+l)^{th}$ element as

$$w_{(m,m+l)} = E[\epsilon(m) \cdot \epsilon^*(m+l)] \tag{2.58}$$

$$w_{R(m,m+l)} = E[\epsilon_R(m) \cdot \epsilon_R(m+l)] \tag{2.59}$$

$$w_{I(m,m+l)} = E[\epsilon_I(m) \cdot \epsilon_I(m+l)] \tag{2.60}$$

$$w_{RI(m,m+l)} = E[\epsilon_R(m) \cdot \epsilon_I(m+l)] \tag{2.61}$$

[6]For the equation error signal model, they become (W_e, W_{e_R}, W_{e_I} and $W_{e_{RI}}$). Accordingly, for the measurement noise signal model, they will be designated by (W_s, W_{s_R}, W_{s_I} and $W_{s_{RI}}$).

where $m = 0, 1, 2, ..., M$, $l = 0, \pm 1, \pm 2, ...$ Then for the signal models mentioned earlier, we have

Lemma 4 (relations among the different covariance matrices) *Under the measurement noise signal model as shown in Figure 2.1 and the modulating function set* (2.51), *if* $n(t)$ *and* $v(t)$ *are mutually independent white Gaussian noises, then the following relations hold true:*

$$w_{s_{R(m,m+l)}} = \frac{1}{2} w_{s_{(m,m+l)}} + \begin{cases} 0 & ; \ m \neq 0 \ or \ l \neq 0 \\ \frac{\sigma_n^2}{2T} a_n^2 + \frac{\sigma_v^2}{2T} b_n^2 & ; \ m = 0 \ and \ l = 0 \end{cases}, \tag{2.62}$$

$$w_{s_{I(m,m+l)}} = \frac{1}{2} w_{s_{(m,m+l)}} - \begin{cases} 0 & ; \ m \neq 0 \ or \ l \neq 0 \\ \frac{\sigma_n^2}{2T} a_n^2 + \frac{\sigma_v^2}{2T} b_n^2 & ; \ m = 0 \ and \ l = 0 \end{cases}, \tag{2.63}$$

where $w_{s_{(m,m+l)}}$ *is defined in Lemma 3, eqn.* (2.46), *and*

$$w_{s_{RI(m,m+l)}} = 0. \tag{2.64}$$

Proof: From definition (2.55)

$$\epsilon_R(m) = \frac{1}{2}[\epsilon(m) + \epsilon^*(m)] \tag{2.65}$$

therefore

$$\begin{aligned} w_{s_{R(m,m+l)}} &= E[\epsilon_R(m) \cdot \epsilon_R(m+l)] \\ &= \frac{1}{4} E\left\{[\epsilon(m) + \epsilon^*(m)] \cdot [\epsilon(m+l) + \epsilon^*(m+l)]\right\} \\ &= \frac{1}{4} E\left\{\epsilon(m)\epsilon(m+l) + \epsilon(m)\epsilon^*(m+l) + \epsilon^*(m)\epsilon(m+l) + \epsilon^*(m)\epsilon^*(m+l)\right\} \end{aligned} \tag{2.66}$$

Here we need first to prove the fact that terms like $E[\epsilon(m)\epsilon(m+l)]$ and $E[\epsilon^*(m)\epsilon^*(m+l)]$ will contribute to equation (2.66) only when $m=0$ and $l=0$ due to the orthogonality of $\{e^{-jk\omega_0 t}, k=0,1,2,...\}$.

Using the condition of mutual independence, equation (2.45), and Corollary 3 in Appendix D,

$$E[\epsilon(m)\epsilon(m+l)] = \underbrace{\frac{\sigma_n^2}{T^2}\sum_{k_1=m}^{n+m}\sum_{k_2=m+l}^{n+m+l}\alpha(k_1,m,\theta_a)\alpha(k_2,m+l,\theta_a)\int_0^T\int_0^T E[n(t_1)n(t_2)]e^{-j[k_1\omega_0 t+k_2\omega_0 t]}dt_1dt_2}_{\Pi_1} + \underbrace{\frac{\sigma_v^2}{T^2}\sum_{k_1=m}^{n+m}\sum_{k_2=m+l}^{n+m+l}\beta(k_1,m,\theta_b)\beta(k_2,m+l,\theta_b)\int_0^T\int_0^T E[v(t_1)v(t_2)]e^{-j[k_1\omega_0 t+k_2\omega_0 t]}dt_1dt_2}_{\Pi_2} \tag{2.67}$$

Note the fact that

$$E[n(t_1)\cdot n(t_2)] = \sigma_n^2\delta_D(t_1-t_2).$$

Then the first term of equation (2.67) can be reduced to

$$\Pi_1 = \frac{\sigma_n^2}{T^2}\sum_{k_1=m}^{n+m}\sum_{k_2=m+l}^{n+m+l}\alpha(k_1,m,\theta_a)\alpha(k_2,m+l,\theta_a)\int_0^T e^{-j(k_1+k_2)\omega_0 t}dt. \tag{2.68}$$

In the above, $m\geq 0$ and $m+l\geq 0$, so $k_1\geq 0$ and $k_2\geq 0$. But due to

$$\int_0^T e^{-j(k_1+k_2)\omega_0 t}dt = \begin{cases} 0 & ; \quad k_1+k_2\neq 0 \\ T & ; \quad k_1+k_2=0 \end{cases}, \tag{2.69}$$

we have

$$\Pi_1 = \begin{cases} 0 & ; \quad k_1+k_2\neq 0 \\ \frac{\sigma_n^2}{T}\alpha^2(0,0,\theta_a) & ; \quad k_1+k_2=0 \end{cases}. \tag{2.70}$$

Condition $k_1+k_2=0$ is equivalent to $m=0$ and $l=0$. Further from the definition

$$\alpha(k,m,\theta_a) = c_{k-m}\left((jk\omega_0)^n,...,(jk\omega_0)^0\right)\begin{pmatrix} 1 \\ -\theta_a \end{pmatrix} \tag{2.71}$$

we have $\alpha(0,0,\theta_a) = -a_n$. Therefore,

$$\Pi_1 = \begin{cases} 0 & ; \quad m \neq 0 \text{ or } l \neq 0 \\ \frac{\sigma_n^2}{T} a_n^2 & ; \quad m = 0 \text{ and } l = 0 \end{cases} . \tag{2.72}$$

Following a similar argument, we have

$$\Pi_2 = \begin{cases} 0 & ; \quad m \neq 0 \text{ or } l \neq 0 \\ \frac{\sigma_v^2}{T} b_n^2 & ; \quad m = 0 \text{ and } l = 0 \end{cases} \tag{2.73}$$

as well. Substituting Π_1 and Π_2 into (2.67)

$$E[\epsilon(m)\epsilon(m+l)] = \begin{cases} 0 & ; \quad m \neq 0 \text{ or } l \neq 0 \\ \frac{\sigma_n^2}{T} a_n^2 + \frac{\sigma_v^2}{T} b_n^2 & ; \quad m = 0 \text{ and } l = 0 \end{cases} \tag{2.74}$$

is proved. This also automatically implies

$$E[\epsilon^*(m)\epsilon^*(m+l)] = \{E[\epsilon(m)\epsilon(m+l)]\}^* = \begin{cases} 0 & ; \quad m \neq 0 \text{ or } l \neq 0 \\ \frac{\sigma_n^2}{T} a_n^2 + \frac{\sigma_v^2}{T} b_n^2 & ; \quad m = 0 \text{ and } l = 0 \end{cases} . \tag{2.75}$$

Using (2.74), (2.75) and Lemma 3, equation (2.66) can be rewritten as

$$\begin{aligned} w_{s_{R(m,m+l)}} &= \frac{1}{2} \cdot \{E[\epsilon(m)\epsilon^*(m+l)] + E[\epsilon(m)\epsilon(m+l)]\} \\ &= \frac{1}{2} w_{s_{(m,m+l)}} + \begin{cases} 0 & ; \quad m \neq 0 \text{ or } l \neq 0 \\ \frac{\sigma_n^2}{2T} a_n^2 + \frac{\sigma_v^2}{2T} b_n^2 & ; \quad m = 0 \text{ and } l = 0 \end{cases} \end{aligned} \tag{2.76}$$

If it is further noted that

$$\epsilon_I(m) = \frac{1}{2j}[\epsilon(m) - \epsilon^*(m)] \tag{2.77}$$

then

$$w_{s_{I(m,m+l)}} = E[\epsilon_I(m)\epsilon_I(m+l)]$$

$$
\begin{aligned}
&= \frac{1}{4j^2} \cdot E\{[\epsilon(m) - \epsilon^*(m)][\epsilon(m+l) - \epsilon^*(m+l)]\} \\
&= \frac{1}{-4} E\{\epsilon(m)\epsilon(m+l) - \epsilon^*(m)\epsilon(m+l) - \epsilon(m)\epsilon^*(m+l) + \epsilon^*(m)\epsilon^*(m+l)\}
\end{aligned}
\tag{2.78}
$$

Similarly considering the results of (2.74), (2.75) and Lemma 3, the above relation is reduced to

$$
\begin{aligned}
w_{s_{I(m,m+l)}} &= \frac{1}{2} \cdot \{E[\epsilon(m)\epsilon^*(m+l)] - E[\epsilon(m)\epsilon(m+l)]\} \\
&= \frac{1}{2} w_{s_{(m,m+l)}} - \begin{cases} 0 & ; \quad m \neq 0 \text{ or } l \neq 0 \\ \frac{\sigma_n^2}{2T} a_n^2 + \frac{\sigma_v^2}{2T} b_n^2 & ; \quad m = 0 \text{ and } l = 0 \end{cases}
\end{aligned}
\tag{2.79}
$$

Hence, relations (2.62) and (2.63) are established. Finally, using identities (2.65), (2.77), (2.74), (2.75) and Lemma 3:

$$
\begin{aligned}
w_{s_{RI(m,m+l)}} &= E[\epsilon_R(m)\epsilon_I(m+l)] \\
&= \frac{1}{4j} \cdot E\{[\epsilon(m) + \epsilon^*(m)] \cdot [\epsilon(m+l) - \epsilon^*(m+l)]\} \\
&= \frac{1}{2} \cdot Im\{w_{s_{(m,m+l)}}\} + \frac{1}{4j} \cdot E[\epsilon(m)\epsilon(m+l)] - \frac{1}{4j} \cdot E[\epsilon^*(m)\epsilon^*(m+l)] \\
&= 0
\end{aligned}
\tag{2.80}
$$

Therefore, the proof of this Lemma is completed. In the form of a matrix, the above Lemma implies:

$$
W_{s_{RI}} = 0, \tag{2.81}
$$

$$
W_{s_R} = \frac{1}{2}(W_s + \eta_s \eta_s^T), \tag{2.82}
$$

$$
W_{s_I} = \frac{1}{2}(W_s - \eta_s \eta_s^T), \tag{2.83}
$$

where the column vector η is defined as

$$\begin{aligned}\eta_s &= \begin{pmatrix} \sqrt{\frac{\sigma_n^2}{T}a_n^2 + \frac{\sigma_v^2}{T}b_n^2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} \sqrt{\frac{\sigma_n^2}{T}a_n^2 + \frac{\sigma_v^2}{T}b_n^2} \\ \mathcal{O}_{M\times 1} \end{pmatrix}. \end{aligned} \tag{2.84}$$

As a matter of fact, Lemma 2 and the equation error signal model could be just deemed as a special case of Lemma 3 and Lemma 4, and correspondingly, η_e should be modified as:

$$\eta_e = \begin{pmatrix} \sqrt{\frac{\sigma^2}{T}} \\ \mathcal{O}_{M\times 1} \end{pmatrix}, \tag{2.85}$$

and

$$W_{e_{RI}} = 0, \tag{2.86}$$

$$W_{e_R} = \frac{1}{2}(W_e + \eta_e\eta_e^T), \tag{2.87}$$

$$W_{e_I} = \frac{1}{2}(W_e - \eta_e\eta_e^T). \tag{2.88}$$

Hence, the following derivations about implementation schemes will be applicable to both stochastic signal models, and subscripts "e" and "s" will be dropped for the general discussion.

2.3.2 Joint Cost Function

Splitting the modulated quantities into real and imaginary parts is equivalent to using both real and imaginary parts of the complex modulating function set (2.4) to modulate the original differential equation model separately. Although it would be

quite awkward, both real and imaginary parts could work independently to estimate the very same parameter vector θ using the weighted least square algorithms with different weights W_R and W_I through minimizing two different cost functions. In order to avoid the potential agony induced by this separation, the possibility of using one joint cost function binding these two parts to produce just one estimate, instead of two, should be explored. Under the assumption that additive noises $n(t)$ and $v(t)$ are mutually independent white Gaussian noise, the error sequences ϵ_R and ϵ_I in (2.56) and (2.57) are uncorrelated as shown in Lemma 4, and then the joint likelihood function is as follows:

$$\begin{aligned} p(\epsilon_R, \epsilon_I \mid \theta) &= p(\epsilon_R \mid \theta) \cdot p(\epsilon_I \mid \theta) \\ &= \frac{1}{(2\pi)^{\frac{M+1}{2}} |W_R|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\epsilon_R^T W_R^{-1} \epsilon_R\right\} \frac{1}{(2\pi)^{\frac{M+1}{2}} |W_I|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}\epsilon_I^T W_I^{-1} \epsilon_I\right\} \\ &= \frac{1}{(2\pi)^{M+1} |W_R|^{\frac{1}{2}} |W_I|^{\frac{1}{2}}} \cdot \exp\left\{-\frac{1}{2} J(\theta)\right\} \end{aligned} \tag{2.89}$$

where

$$\begin{aligned} J(\theta) &= \epsilon_R^T W_R^{-1} \epsilon_R + \epsilon_I^T W_I^{-1} \epsilon_I \\ &= (Y_R - \Gamma_R \theta)^T W_R^{-1} (Y_R - \Gamma_R \theta) + (Y_I - \Gamma_I \theta)^T W_I^{-1} (Y_I - \Gamma_I \theta) \\ &\geq 0. \end{aligned}$$

The log-likelihood function $\mathcal{L}(\theta)$ can be used and

$$\begin{aligned} \mathcal{L}(\theta) &= \ln p(\epsilon_R, \epsilon_I | \theta) \\ &= -(M+1) \ln 2\pi - \frac{1}{2}[\ln |W_R| + \ln |W_I|] - \frac{1}{2} J(\theta). \end{aligned} \tag{2.90}$$

If the covariance matrices W_R and W_I are **known**, maximizing $\mathcal{L}(\theta)$ for the maximum likelihood estimate is equivalent to minimizing the quadratic form of the function $J(\theta)$. For this reason,

$$J(\theta) = (Y_R - \Gamma_R\theta)^T W_R^{-1}(Y_R - \Gamma_R\theta) + (Y_I - \Gamma_I\theta)^T W_I^{-1}(Y_I - \Gamma_I\theta) \tag{2.91}$$

has been selected as the joint cost function for the rest of our studies on SISO systems. The combined or joint estimate desired should be

$$\hat{\theta} = arg\min_{\theta} J(\theta). \tag{2.92}$$

From the necessary condition of minimization: $\frac{\partial J}{\partial \theta} = 0$, we have

$$\begin{aligned} \frac{\partial J}{\partial \theta} &= \Gamma_R^T W_R^{-1}(Y_R - \Gamma_R\theta) + \Gamma_I^T W_I^{-1}(Y_I - \Gamma_I\theta) \\ &= (\Gamma_R^T W_R^{-1} Y_R + \Gamma_I^T W_I^{-1} Y_I) - (\Gamma_R^T W_R^{-1}\Gamma_R + \Gamma_I^T W_I^{-1}\Gamma_I)\theta \\ &= 0 \end{aligned} \tag{2.93}$$

which implies

$$\hat{\theta} = \{\Gamma_R^T W_R^{-1}\Gamma_R + \Gamma_I^T W_I^{-1}\Gamma_I\}^{-1} \cdot \{\Gamma_R^T W_R^{-1} Y_R + \Gamma_I^T W_I^{-1} Y_I\}. \tag{2.94}$$

In order to have a WLS form like (2.35), further introduce the following combined notations:

$$\Gamma_C = \begin{pmatrix} \Gamma_R \\ \Gamma_I \end{pmatrix} \tag{2.95}$$

$$Y_C = \begin{pmatrix} Y_R \\ Y_I \end{pmatrix} \tag{2.96}$$

$$W_C = \begin{pmatrix} W_R & 0 \\ 0 & W_I \end{pmatrix}. \tag{2.97}$$

If we denote

$$W_C^{-1} = \begin{pmatrix} W_R^{-1} & 0 \\ 0 & W_I^{-1} \end{pmatrix}$$

then it is straightforward to show that

$$\begin{aligned} \hat{\theta} &= \{\Gamma_R^T W_R^{-1} \Gamma_R + \Gamma_I^T W_I^{-1} \Gamma_I\}^{-1} \cdot \{\Gamma_R^T W_R^{-1} Y_R + \Gamma_I^T W_I^{-1} Y_I\} \\ &= (\Gamma_C^T W_C^{-1} \Gamma_C)^{-1} \Gamma_C^T W_C^{-1} Y_C. \end{aligned} \tag{2.98}$$

This result shows that by minimizing the joint cost function (2.91) we still can have a combined WLS estimate $\hat{\theta}$ through the combined regressor, regressand and weighting. This also has provided an efficient way to utilize the information carried in both the real and imaginary parts of modulated quantities.

Further, we have the following observations about implementing the above scheme:

1. For the regular least squares algorithm, $W_C = I$, then

$$\hat{\theta}_{LS} = (\Gamma_C^T \Gamma_C)^{-1} \Gamma_C^T Y_C.$$

2. For the equation error signal model and measurement noise signal model of Figure 2.1, two matrix inverses W_R^{-1} and W_I^{-1} are needed for W_C^{-1}. However, utilizing the matrix inversion lemma [2]:

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}, \tag{2.99}$$

and letting $A = W$, $B = \eta$, $C = I$ and $D = \eta^T$, the special form (2.82) of W_R becomes

$$W_R^{-1} = 2(W + \eta\eta^T)^{-1} = 2(W^{-1} - W^{-1}\eta(1 + \eta^T W^{-1}\eta)^{-1}\eta^T W^{-1}). \tag{2.100}$$

Denoting $\delta_s = \sqrt{\frac{\sigma_n^2}{T}a_n^2 + \frac{\sigma_v^2}{T}b_n^2}$ for the measurement noise signal model (or $\delta_e = \sqrt{\frac{\sigma^2}{T}}$ for the equation error signal model) and $\Lambda = \eta\eta^T W^{-1}$ (with a similar designation Λ_s and Λ_e), and partitioning

$$\eta = \begin{pmatrix} \delta \\ \mathcal{O}_{M\times 1} \end{pmatrix} \qquad W^{-1} = \begin{pmatrix} w_{I11} & W_{I12} \\ W_{I21} & W_{I22} \end{pmatrix}, \tag{2.101}$$

we have

$$\eta^T W^{-1}\eta = \delta^2 w_{I11} \tag{2.102}$$

$$\begin{aligned} \Lambda &= \eta\eta^T W^{-1} \\ &= \begin{pmatrix} \delta^2 w_{I11} & \delta^2 W_{I12} \\ \mathcal{O}_{M\times 1} & \mathcal{O}_{M\times M} \end{pmatrix}. \end{aligned} \tag{2.103}$$

It is straightforward to show

$$W_R^{-1} = 2W^{-1}(I - \frac{\Lambda}{1+\delta^2 w_{I11}}). \tag{2.104}$$

Similarly, we have

$$W_I^{-1} = 2W^{-1}(I + \frac{\Lambda}{1-\delta^2 w_{I11}}). \tag{2.105}$$

Equations (2.104) and (2.105) indicate that *only one* matrix inverse W^{-1} is needed at each iteration.

3. One very important remark that should be reiterated here is the fact that if W_R and W_I are not known beforehand, while they may be explicitly expressed as a function of the parameter θ such as shown in Lemma 3 and Lemma 4 for the measurement noise signal model, the AWLS/MFT estimate stated in Algorithm 2 does not lead to the **exact** maximum likelihood estimate. But

when the estimated $\hat{\theta}$ through minimizing $J(\theta)$ is not far away from the true θ, the estimated $\hat{W}$ should also be very close to the true one as well. This is why the claim that AWLS/MFT is just an approximated maximum likelihood estimate has been declared. In order to have a true maximum likelihood estimate, the maximization should be applied directly to the log-likelihood function (2.90). The terms like $\ln|W_R|$ and $\ln|W_I|$ could truely make the computation become formidable. In this aspect, $J(\theta)$ is much more attractive.

4. For computational simplicity, one may simply use a single (approximate) real W as the weighting for both W_R and W_I in each iteration and neglect the true updating forms (2.104) and (2.105) caused by the tiny difference between W_R and W_I. As one revealing example to acertain the cost of this simplicity, let us identify the following second order system:

$$H(s) = \frac{8}{s^2 + 4s + 10}. \tag{2.106}$$

The AWLS/MFT Algorithm 2 will be utilized with and without forms (2.104) and (2.105) using 100 Monte Carlo simulation runs for each case at each of several additive noise levels. In order to see the relative difference one to another on both the mean and standard deviation, define a percent error measure by:

$$\Delta = \frac{\|\xi_e - \xi_a\|_2}{\|\xi_e\|_2} \cdot 100\%. \tag{2.107}$$

Here ξ_e corresponds to either the mean or the standard deviation values for each parameter obtained *with* the exact weightings as defined in (2.104) and

(2.105), and ξ_a corresponds to either the mean or the standard deviation values for each parameter obtained *without* the exact weightings as defined in (2.104) and (2.105). The results are summarized in Table 2.1, from which we can see

true para	8		4		10		
	mean	std	mean	std	mean	std	NSR
EXACT	7.9920	0.08105	4.0025	0.04378	9.9970	0.07880	5%
APPROX	7.9862	0.08700	3.9997	0.04579	9.9930	0.08265	
EXACT	7.9661	0.15862	3.9795	0.09232	9.9710	0.1424	10%
APPROX	7.9621	0.16976	3.9777	0.09795	9.9688	0.1513	
EXACT	7.9033	0.30242	3.9500	0.17236	9.9160	0.29532	20%
APPROX	7.9032	0.33463	3.9503	0.18710	9.9164	0.31104	
EXACT	7.5696	0.61485	3.7424	0.33769	9.6179	0.54303	40%
APPROX	7.5373	0.61791	3.7259	0.34017	9.5889	0.55101	
EXACT	6.6254	0.95782	3.1916	0.54162	8.7639	0.78983	80%
APPROX	6.5373	1.00141	3.1437	0.55088	8.6856	0.79628	
Δ	0.55%	4.68%	0.60%	2.78%	0.39%	2.08%	

100 Monte Carlo runs:
APPROX means using Ws as both Ws_R and Ws_I without using (2.104) and (2.105).
EXACT means using exact the Ws_R and Ws_I as defined in (2.104) and (2.105).

Table 2.1: Comparison between using exact and approximate weighting matrices

that the increased accuracy in using (2.104) and (2.105) has only a slight edge, i.e., 0.6% in mean and 4.68% in std, over the case without using (2.104) and (2.105). As for the speed of convergence and computational time concerns, they do not exhibit any difference. In the rest of this thesis, the results in applying AWLS/MFT are obtained without using (2.104) and (2.105).

2.3.3 A Simple Recursive Algorithm for Matrix Inversion

We have seen that one of the major computational burdens is the inversion of an $(M+1) \times (M+1)$ weighting matrix needed in implementing the WLS/MFT and AWLS/MFT algorithms. Numerical experiments show that when the order of the modulating function set or the model goes higher than 10, the MATLAB's matrix inverse routine which uses "matrix division" or singular value decomposition would fail to provide usable answers. Part of the reason is that those routines are not specifically written to deal with banded symmetric positive definite matrices like our weightings. The round-off errors could accumulate very fast or the matrix could be badly scaled, especially when M gets large. With these particular sparse matrix structures in mind, we hope that we can contrive some algorithm which eventually avoids direct matrix inversion and also can utilize the sparse structure of the weighting matrix to improve the numerical accuracy and efficiency.

If the upper-left $(k+1) \times (k+1)$ sub-matrix of the weighting W is denoted by W_{k+1}, we need first to answer the following question:

> Provided that W_k^{-1} is known, is it possible to compute W_{k+1}^{-1} from W_k^{-1} without employing a multidimensional matrix inversion ?

Partition the $(k+1) \times (k+1)$ sub-weighting matrix W_{k+1}

$$W_{k+1} = \begin{pmatrix} W_k & B_k \\ B_k^T & a_k \end{pmatrix} \tag{2.108}$$

where B_k is $k \times 1$ column vector and a_k is the $(k+1)$-th diagonal element of the

weighting matrix W. Due to the symmetric positive definite property, the inverse of W_{k+1} exists and is also symmetric positive definite. If we denote

$$W_{k+1}^{-1} = \begin{pmatrix} V_k & C_k \\ C_k^T & d_k \end{pmatrix} \tag{2.109}$$

where V_k is a $k \times k$ matrix, C_k is $k \times 1$ column vector and d_k is the $(k+1)$-th diagonal element of W^{-1}, it follows from the condition

$$\begin{aligned} W_{k+1} \cdot W_{k+1}^{-1} &= \begin{pmatrix} W_k & B_k \\ B_k^T & a_k \end{pmatrix} \cdot \begin{pmatrix} V_k & C_k \\ C_k^T & d_k \end{pmatrix} \\ &= \begin{pmatrix} W_k V_k + B_k C_k^T & W_k C_k + B_k d_k \\ B_k^T V_k + a_k C_k^T & B_k^T C_k + a_k d_k \end{pmatrix} \\ &= \begin{pmatrix} I_k & 0 \\ 0 & 1 \end{pmatrix} \end{aligned}$$

that we have four equations:

$$B_k^T C_k + a_k d_k = 1 \tag{2.110}$$

$$W_k C_k + B_k d_k = 0 \tag{2.111}$$

$$B_k^T V_k + a_k C_k^T = 0 \tag{2.112}$$

$$W_k V_k + B_k C_k^T = I_k. \tag{2.113}$$

We wish to solve for the three unknowns (V_k, C_k, d_k) with $(W_k, W_k^{-1}, B_k, a_k)$ as knowns. From equation (2.113) and (2.110) we have

$$V_k = W_k^{-1} \cdot (I_k - B_k C_k^T) \tag{2.114}$$

$$d_k = \frac{1 - B_k^T C_k}{a_k}. \tag{2.115}$$

Substituting (2.115) into (2.111):

$$W_k C_k + B_k \frac{1 - B_k^T C_k}{a_k} = 0$$

$$(W_k - \frac{B_k B_K^T}{a_K})C_k = -\frac{B_k}{a_k}.$$

So

$$C_k = -[W_k - \frac{B_k B_k^T}{a_k}]^{-1} \cdot \frac{B_k}{a_k} \tag{2.116}$$

This expression for C_k still requires computing a $k \times k$ matrix inverse $[W_k - \frac{B_k B_k^T}{a_k}]^{-1}$ which is not desired. But from the well known Matrix Inversion Lemma,

$$\begin{aligned} C_k &= -[W_k - \frac{B_k B_k^T}{a_k}]^{-1} \cdot \frac{B_k}{a_k} \\ &= -[W_k^{-1} - W_k^{-1} B_k (B_k^T W_k^{-1} B_k - a_k)^{-1} B_k^T W_k^{-1}] \frac{B_k}{a_k} \\ &= -[I_k + \frac{W_k^{-1} B_k B_k^T}{a_k - B_k^T W_k^{-1} B_k}] \frac{W_k^{-1} B_k}{a_k} \\ &= -\frac{W_k^{-1} B_k}{a_k}[1 - \frac{B_k^T W_k^{-1} B_k}{B_k^T W_k^{-1} B_k - a_k}] \\ &= \frac{W_k^{-1} B_k}{B_k^T W_k^{-1} B_k - a_k}. \end{aligned} \tag{2.117}$$

Putting this back into equation (2.115) and (2.114) we obtain

$$d_k = -\frac{1}{B_k^T W_k^{-1} B_k - a_k} \tag{2.118}$$

$$V_k = W_k^{-1}[I_k - \frac{B_k B_k^T W_k^{-1}}{B_k^T W_k^{-1} B_k - a_k}] \tag{2.119}$$

Further, by defining a column vector Λ_k by $\Lambda_k = W_k^{-1} B_k$ and a scalar $\tau_k = B_k^T W_k^{-1} B_k$, and combining (2.117) with (2.119) and (2.118), we have the following algorithm for any positive definite matrix W.

Algorithm 4 (recursive matrix inversion for general weighting) *For a known $M \times M$ positive definite matrix W with main diagonal elements $\{a_k\}$; $k = 0, 1, ..., M-1$, its inverse can be computed in the following way:*

1. *Initialize index* $k = 1$ *and* $W_1^{-1} = \frac{1}{a_0}$.

2. *Increment k=k+1 and check if* $k > M$ *or not ? Yes: stop; No: continue.*

3. *Obtain* B_k *and* a_k *directly from partitioning* W_k *as in the right side of* (2.108).

4. *Compute*

$$\begin{aligned} \Lambda_k &= W_k^{-1} \cdot B_k \\ \tau_k &= B_k^T \cdot \Lambda_k \\ C_k &= \frac{\Lambda_k}{\tau_k - a_k} \\ V_k &= W_k^{-1} - \frac{\Lambda_k \Lambda_k^T}{\tau_k - a_k} \\ d_k &= -\frac{1}{\tau_k - a_k}. \end{aligned}$$

5. *Form*

$$W_{k+1}^{-1} = \begin{pmatrix} V_k & C_k \\ C_k^T & d_k \end{pmatrix}.$$

6. *Go to step 2.*

Therefore, the question of recursive updating W_{k+1}^{-1} from W_k^{-1} has been answered. Following up, we will further take the sparse structure of the weighting matrices into account, so that the above algorithm can be made more efficient to compute Λ_k and τ_k at each recursion. If the order of the modulating function set is n, from Lemma 2 and Lemma 3 the bandwidth of the covariance would be n. Therefore, except for $k \leq n$, the column $k \times 1$ matrix can be partitioned as

$$B_k = \begin{pmatrix} \mathcal{O}^{(k-n)\times 1} \\ \bar{B}_k^{n\times 1} \end{pmatrix} \tag{2.120}$$

where $\mathcal{O}^{(k-n)\times 1}$ is a $(k-n)\times 1$ zero column vector and $\bar{B}_k^{n\times 1}$ is a $n\times 1$ matrix. Correspondingly

$$W_k^{-1} = \begin{pmatrix} R_{k,11}^{(k-n)\times(k-n)} & R_{k,12}^{(k-n)\times n} \\ (R_{k,12}^{(k-n)\times n})^T & R_{k,22}^{n\times n} \end{pmatrix}. \tag{2.121}$$

Then it is straightforward to obtain that

$$\Lambda_k = \begin{pmatrix} R_{k,12}^{(k-n)\times n}\cdot \bar{B}_k^{n\times 1} \\ R_{k,22}^{n\times n}\cdot \bar{B}_k^{n\times 1} \end{pmatrix} \tag{2.122}$$

$$\tau_k = (\bar{B}_k^{n\times 1})^T R_{k,22}^{n\times n}\bar{B}_k^{n\times 1} \tag{2.123}$$

Armed with these two definitions and Algorithm 4 we are ready to introduce the recursive algorithm for a banded symmetric positive definite matrix.

Algorithm 5 (recursive matrix inversion for banded weighting) *For a known $M\times M$ banded symmetric positive definite matrix W with bandwidth n and main diagonal elements $\{a_k\}; k=0,1,...,M-1$, its inverse can be computed in the following way:*

1. *Initialize index $k=1$ and $W_1^{-1}=\frac{1}{a_0}$.*

2. *Increment k=k+1 and check if $k>M$ or not ? Yes: stop; No: continue.*

3. *Obtain B_k and a_k directly from partitioning W_k as in the right side of* (2.108) *and* (2.120).

4. *Partition W_k and B_k as* (2.121) *and* (2.120) *and compute*

$$\Lambda_k = \begin{pmatrix} R_{k,12}^{(k-n)\times n}\cdot \bar{B}_k^{n\times 1} \\ R_{k,22}^{n\times n}\cdot \bar{B}_k^{n\times 1} \end{pmatrix}$$

$$
\begin{aligned}
\tau_k &= (\bar{B}_k^{n\times 1})^T R_{k,22}^{n\times n} \bar{B}_k^{n\times 1} \\
C_k &= \frac{\Lambda_k}{\tau_k - a_k} \\
V_k &= W_k^{-1} - \frac{\Lambda_k \Lambda_k^T}{\tau_k - a_k} \\
d_k &= -\frac{1}{\tau_k - a_k}.
\end{aligned}
$$

5. *Form*

$$
W_{k+1}^{-1} = \begin{pmatrix} V_k & C_k \\ C_k^T & d_k \end{pmatrix}.
$$

6. *Go to step 2.*

Remarks on the above algorithm:

1. As only the last n columns of W_k^{-1} are involved, the maximum inner product dimension is n instead of k. When $k >> n$, this is very helpful for the depression of accumulation errors. Computationally, kn flops for Λ_k and n^2 flops for τ_k are required to update at each recursion.

2. The most computationally-demanding term in updating V_k is $\Lambda_k \Lambda_k^T$; however, it only involves the product operations among the elements of the column vector Λ_k. Hence, it does not contribute to accumulation errors at each recursion.

3. The total flops required is of order $O(M^3)$, which is the same as LU decomposition and Gauss-Jordan elimination methods.

4. When W is Toeplitz, i.e., $W = W_e$, we have two choices:

(a) Still use Algorithm 5.

(b) Employ Trench's algorithm (see pp 132 of [9]) which only needs $O(M^2)$ flops. The Trench Algorithm requires the column vector C_M and d_M to be obtained first from Durbin's algorithm which is of $O(M^2)$ flops. Note in Algorithm 5 that if we drop the V_k updating, we also can obtain C_M and d_M in $O(M^2)$ flops which is as efficient as Durbin's algorithm. Then at least we can use the above algorithm to first obtain C_M and d_M.

5. The bottom line is that numerically the above algorithm is much more robust. It has successfully inverted a $M = 1024$, $n = 12$, Toeplitz matrix while the routines in MATLAB failed. For AWLS/MFT, it is as efficient as any other inversion algorithm.

6. One other by-product of the above algorithm is that it facilitates writing a recursive weighted least squares algorithm, which might not be necessary in MFT, but it may be of value to other sequentially correlated data analyses.

2.4 Comparing LS, WLS and AWLS with PEM

The second order system: $\ddot{y}(t)+3\dot{y}(t)+8y(t) = 5u(t)$, where $0 \leq t \leq T$ and $T = 10sec$, was used to evaluate and compare the performance of the LS/MFT, WLS/MFT and AWLS/MFT algorithms and to compare with a commercially available PEM (prediction error method) algorithm [20] in MATLAB written by L. Ljung [21]. Two

hundred Monte Carlo runs were made at each of several noise-to-signal ratios for additive white output noise corrupted data. The input signal was $u(t) = \sin(t^2/5)$, $t \in [0, 10]$ secs for each run and the sampling rate is fixed as 25.6Hz. The output $y(t)$ is a combination of the simulated output using LSIM() of MATLAB and the white Gaussian random noise sequence generated by RANDN(), i.e.,

$$y(t) = \text{LSIM}(A, B, C, D, u, t, X0) + n(t)$$

where $[A, B, C, D] = \text{TF2SS}(5, [1, 3, 5])$, $X0$ is the initial condition, and $n(t) = k * \text{RANDN}(256, 1)$ is the additive noise with the scale factor k determining the noise level. In order to have a fair and accurate comparison, every noisy input/output realization pair has been forced to run through all four algorithms in each Monte-Carlo trial. The noise-to-signal ratio (NSR) is defined as

$$\text{NSR} = \frac{\|n(t)\|_2}{\|y(t)\|_2} \cdot 100\% \tag{2.124}$$

which characterizes the percent additive noise on the output. As for a true parameter ϱ_0 and its estimate ϱ (with standard deviation σ), a normalized bias and standard deviation are formed as

$$\text{Normalized Bias} = |\frac{\varrho - \varrho_0}{\varrho_0}| \cdot 100\% \tag{2.125}$$

$$\text{Normalized STD} = |\frac{\sigma}{\varrho}| \cdot 100\% \tag{2.126}$$

These will be used to measure the accuracy of the different algorithms. For the above specific system, its step response will take about 4 seconds to reach steady state and,

therefore, for the total 10 seconds of data, the initial condition $X0$ and the input could both play an important role. In order to have a better picture of the impact of initial conditions on the estimation, two cases with and without randomized $X0$ have been carried out in the following simulation studies.

2.4.1 With $X0$ Fixed as $(0,0)'$

In this case we assume that the initial conditions are always known as $X0 = (0,0)'$, so as not to treat the $X0$ as an unknown in PEM. In each Monte Carlo run of the PEM, the initial guess of the parameters is set favorably to the true values as well as giving it the true value $X0 = (0,0)'$. Under this relatively ideal setting for PEM, the simulation results are summarized in the Table 2.2 $\sim$ 2.5 and Figure 2.2 from which we have the following observations:

1. Although the PEM has a smaller variance than LS/MFT at most noise levels, especially in the lower noise level cases[7], PEM does have greater variance than both WLS/MFT and AWLS/MFT algorithms at all the additive noise levels. The variances in WLS/MFT or AWLS/MFT have been reduced to about one-third the variance of LS/MFT. Between the standard deviations of WLS/MFT and AWLS/MFT, the latter has a slight edge over the former only at very large additive noise levels.

[7]Fullerton, A. Jr. revealed this fact from his early simulation studies [8]. Our craving of further curbing this quantity triggered our studies on the WLS/MFT and AWLS/MFT algorithms.

200 Monte Carlo runs for PEM without estimating X(0), Fb=0.4Hz, fixed initial X(0)=(0,0)'				
true parameters	8	3	5	$\|y(t)\|_2$=6.472
mean	8.509	3.151	5.320	$\|e(t)\|_2$=0.822
variance	0.0171	0.0069	0.0153	NSR=12.7%
mean	8.504	3.155	5.309	$\|e(t)\|_2$=1.664
variance	0.0775	0.0294	0.0625	NSR=25.7%
mean	8.509	3.146	5.328	$\|e(t)\|_2$=3.330
variance	0.2695	0.1119	0.2435	NSR=51.5%
mean	8.609	3.201	5.417	$\|e(t)\|_2$=4.938
variance	0.6306	0.2530	0.5162	NSR=76.3%
mean	8.599	3.181	5.412	$\|e(t)\|_2$=6.660
variance	1.3089	0.5470	1.1467	NSR=102.9%

Table 2.2: PEM with fixed $X(0) = (0,0)'$ and initial guess at true values.

200 Monte Carlo runs for LS/MFT algorithm, Fb=0.4Hz, fixed initial X(0)=(0,0)'				
true parameters	8	3	5	$\|y(t)\|_2$=6.472
mean	7.991	2.988	4.983	$\|e(t)\|_2$=0.822
variance	0.0294	0.0126	0.0290	NSR=12.7%
mean	7.961	2.960	4.947	$\|e(t)\|_2$=1.664
variance	0.1350	0.0557	0.1283	NSR=25.7%
mean	7.886	2.872	4.856	$\|e(t)\|_2$=3.330
variance	0.5144	0.1907	0.4499	NSR=51.5%
mean	7.648	2.698	4.614	$\|e(t)\|_2$=4.938
variance	0.8119	0.3434	0.7714	NSR=76.3%
mean	7.077	2.347	4.111	$\|e(t)\|_2$=6.660
variance	1.1768	0.4449	1.2680	NSR=102.9%

Table 2.3: LS/MFT with fixed $X(0) = (0,0)'$ and $F_b = 0.4$Hz.

200 Monte Carlo runs for WLS/MFT algorithm, Fb=0.4Hz, fixed initial X(0)=(0,0)'				
true parameters	8	3	5	$\|y(t)\|_2$=6.472
mean	7.995	2.989	4.985	$\|e(t)\|_2$=0.822
variance	0.0135	0.0049	0.0103	NSR=12.7%
mean	7.977	2.967	4.955	$\|e(t)\|_2$=1.664
variance	0.0580	0.0228	0.0443	NSR=25.7%
mean	7.859	2.850	4.808	$\|e(t)\|_2$=3.330
variance	0.2129	0.0877	0.1649	NSR=51.5%
mean	7.696	2.665	4.582	$\|e(t)\|_2$=4.938
variance	0.4740	0.1731	0.3920	NSR=76.3%
mean	7.306	2.405	4.253	$\|e(t)\|_2$=6.660
variance	0.4677	0.1779	0.3879	NSR=102.9%

Table 2.4: WLS/MFT with fixed $X(0) = (0,0)'$ and $F_b = 0.4$Hz.

200 Monte Carlo runs for AWLS/MFT algorithm, Fb=0.4Hz, fixed initial X(0)=(0,0)'				
true parameters	8	3	5	$\|y(t)\|_2$=6.472
mean	7.994	2.990	4.986	$\|e(t)\|_2$=0.822
variance	0.0125	0.0045	0.0098	NSR=12.7%
mean	7.974	2.972	4.959	$\|e(t)\|_2$=1.664
variance	0.0542	0.0212	0.0428	NSR=25.7%
mean	7.863	2.870	4.833	$\|e(t)\|_2$=3.330
variance	0.2052	0.0865	0.1620	NSR=51.5%
mean	7.703	2.718	4.630	$\|e(t)\|_2$=4.938
variance	0.4291	0.1678	0.3523	NSR=76.3%
mean	7.338	2.477	4.315	$\|e(t)\|_2$=6.660
variance	0.4079	0.1610	0.3552	NSR=102.9%

Table 2.5: AWLS/MFT with fixed $X(0) = (0,0)'$ and $F_b = 0.4$Hz.

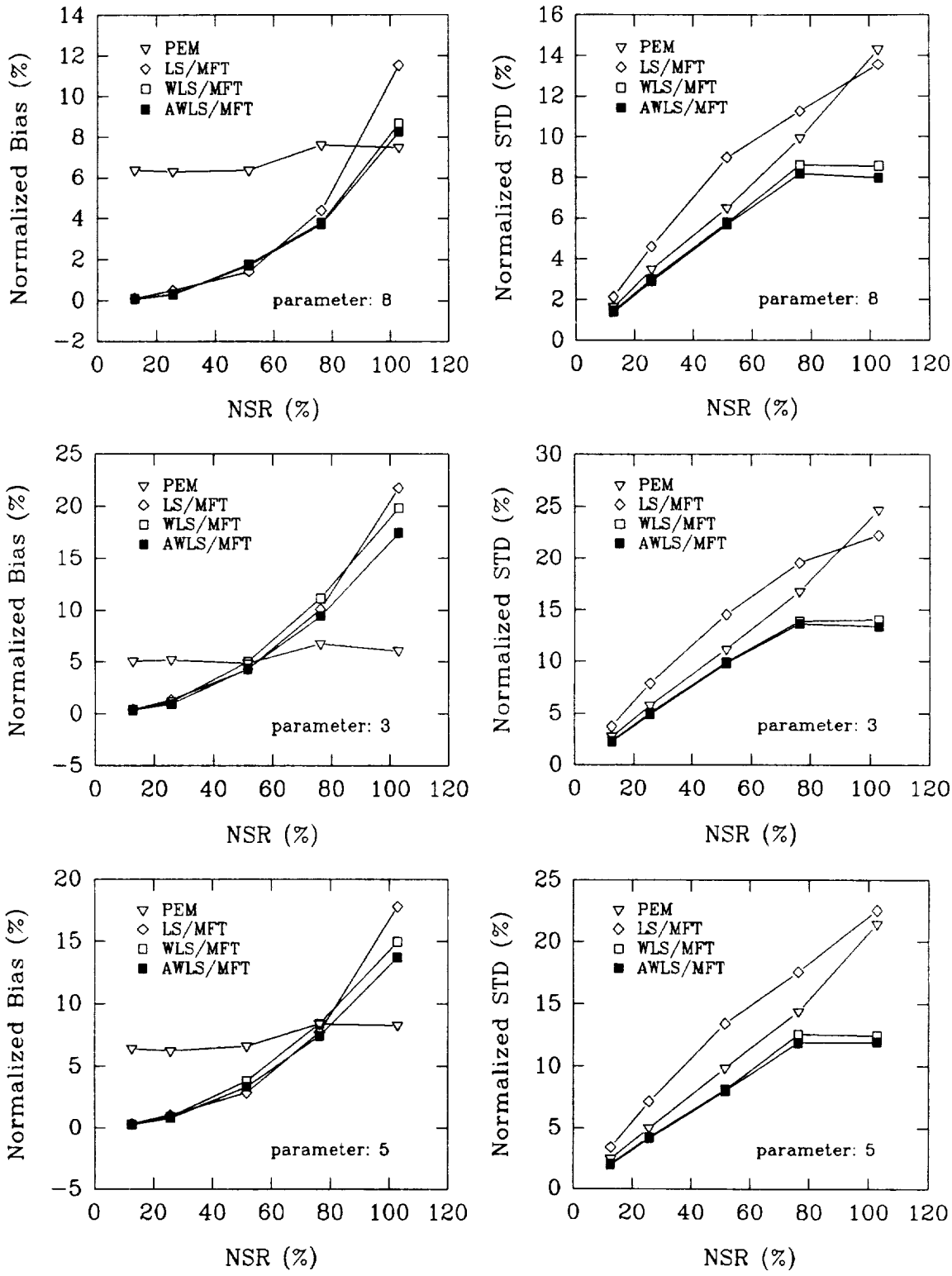

Figure 2.2: Normalized Bias and STD plots with $X(0) = (0, 0)'$

2. As for the bias concern, LS/MFT, WLS/MFT and AWLS/MFT have obviously smaller bias than PEM in the lower noise cases, i.e., $NSR < 56\%$. Among LS/MFT, WLS/MFT and AWLS/MFT, the bias of AWLS/MFT is smaller than that of LS/MFT at the high noise end.

Overall, under this extremely ideal setup favorable to the PEM, WLS/MFT and AWLS/MFT have performed significantly better than PEM within the moderate additive noise range in the sense of variance and bias.

2.4.2 With Randomized $X0$

In this case, each realization of an input/output pair is implemented with randomized initial conditions $X0$. We assume that the initial conditions of the system are unknown beforehand, so that $X0$ has to be estimated in the PEM algorithm while there is no difference to the MFT algorithms. For each Monte Carlo run of PEM, the initial guess of parameters was still set to the true values, but with a randomized initial guess of $X0$ which has to be estimated by the PEM in the end. Under this relatively thorny condition for PEM, the simulation results are summarized in the Table 2.6 $\sim$ 2.9 and Figure 2.3 from which we can make the following remarks:

1. PEM has not only failed to achieve a smaller variance than LS/MFT, but also exhibits unreliability with its frantic-looking mean values. Meanwhile more importantly, there were almost no noticeable effects on the results of the LS/MFT, WLS/MFT and AWLS/MFT algorithms at all noise levels. This kind of robust-

200 Monte Carlo runs for PEM while estimating X(0) with randomized initial X(0)				
true parameters	8	3	5	$\|y(t)\|_2$=6.472
mean	9.273	3.561	5.901	$\|e(t)\|_2$=0.822
variance	0.6927	0.3744	0.7540	NSR=12.7%
mean	10.144	4.291	6.904	$\|e(t)\|_2$=1.664
variance	2.9970	1.1953	2.4566	NSR=25.7%
mean	9.102	3.301	5.610	$\|e(t)\|_2$=3.330
variance	3.4521	1.5663	2.9923	NSR=51.5%
mean	8.927	3.201	5.508	$\|e(t)\|_2$=4.938
variance	4.4852	1.1124	2.8240	NSR=76.3%
mean	8.608	3.145	5.339	$\|e(t)\|_2$=6.660
variance	5.2385	0.8092	2.6997	NSR=102.9%

Table 2.6: PEM with randomized $X(0)$ and initial parameter guess at true values.

200 Monte Carlo runs for LS/MFT algorithm with Fb=0.4Hz and randomized initial X(0)				
true parameters	8	3	5	$\|y(t)\|_2$=6.472
mean	8.008	3.000	5.002	$\|e(t)\|_2$=0.822
variance	0.0225	0.0081	0.0201	NSR=12.7%
mean	7.948	2.959	4.942	$\|e(t)\|_2$=1.664
variance	0.0923	0.0334	0.0932	NSR=25.7%
mean	7.790	2.865	4.787	$\|e(t)\|_2$=3.330
variance	0.3036	0.1131	0.2717	NSR=51.5%
mean	7.488	2.708	4.577	$\|e(t)\|_2$=4.938
variance	0.5972	0.2411	0.6038	NSR=76.3%
mean	7.241	2.469	4.243	$\|e(t)\|_2$=6.660
variance	1.0623	0.4029	1.0296	NSR=102.9%

Table 2.7: LS/MFT with randomized $X(0)'$ and $F_b = 0.4$Hz.

200 Monte Carlo runs for WLS/MFT algorithm with Fb=0.4Hz and randomized initial X(0)				
true parameters	8	3	5	$\|y(t)\|_2$=6.472
mean	7.981	3.005	4.996	$\|e(t)\|_2$=0.822
variance	0.0125	0.0036	0.0096	NSR=12.7%
mean	7.933	2.977	4.950	$\|e(t)\|_2$=1.664
variance	0.0392	0.0125	0.0302	NSR=25.7%
mean	7.901	2.965	4.919	$\|e(t)\|_2$=3.330
variance	0.1304	0.0471	0.0996	NSR=51.5%
mean	7.683	2.834	4.739	$\|e(t)\|_2$=4.938
variance	0.3013	0.0936	0.2541	NSR=76.3%
mean	7.529	2.664	4.554	$\|e(t)\|_2$=6.660
variance	0.5969	0.1792	0.4907	NSR=102.9%

Table 2.8: WLS/MFT with randomized $X(0)'$ and $F_b = 0.4$Hz.

200 Monte Carlo runs for AWLS/MFT algorithm with Fb=0.4Hz and randomized initial X(0)				
true parameters	8	3	5	$\|y(t)\|_2$=6.472
mean	7.989	3.006	4.999	$\|e(t)\|_2$=0.822
variance	0.0111	0.0034	0.0093	NSR=12.7%
mean	7.941	2.981	4.955	$\|e(t)\|_2$=1.664
variance	0.0347	0.0109	0.0273	NSR=25.7%
mean	7.890	2.960	4.912	$\|e(t)\|_2$=3.330
variance	0.1229	0.0442	0.0929	NSR=51.5%
mean	7.668	2.850	4.748	$\|e(t)\|_2$=4.938
variance	0.2725	0.0857	0.2313	NSR=76.3%
mean	7.497	2.673	4.538	$\|e(t)\|_2$=6.660
variance	0.5456	0.1621	0.4479	NSR=102.9%

Table 2.9: AWLS/MFT with randomized $X(0)'$ and $F_b = 0.4$Hz.

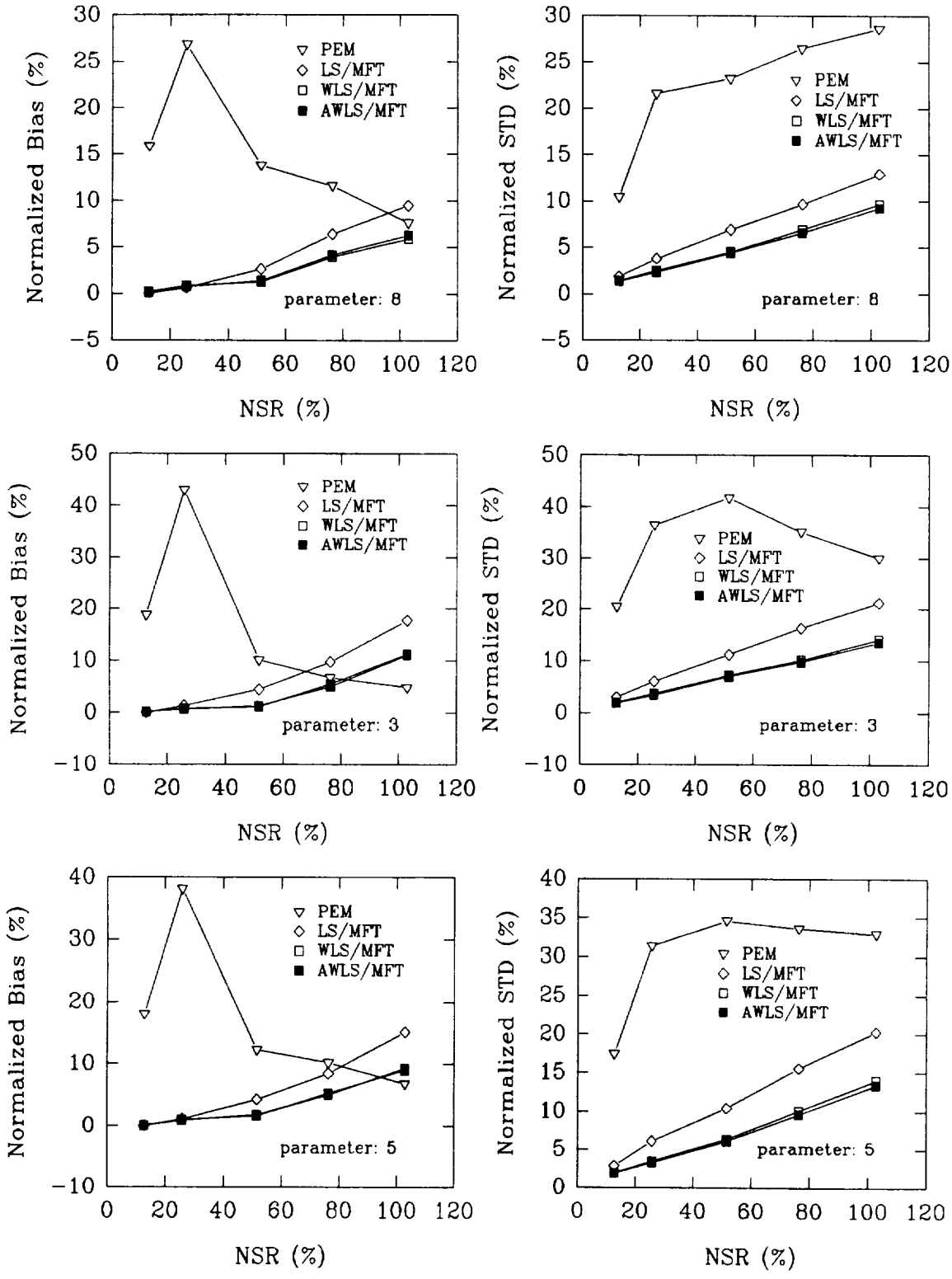

Figure 2.3: Normalized Bias and STD plots with randomized $X(0)$

ness to the randomized initial conditions should not be a surprise if we bear in mind the fact that the MFT itself was originated in a way that the estimation of initial conditions can be totally avoided through the modulating process. Hence, even though almost half of the system response is composed with the transient process, the LS/MFT, WLS/MFT and AWLS/MFT algorithms have not been thwarted at all.

2. In this case, PEM also consumed far more computing time than all the MFT algorithms combined, partly due to the two more unknowns introduced by the initial conditions.

3. Among the MFT algorithms, the WLS/MFT and AWLS/MFT, again manifested improvement through a lower bias and standard deviation.

We have noticed the relatively large bias in the PEM algorithm even in the low noise cases from the above simulation studies. This could be attributed to the fact that the PEM was developed in a discrete time framework. Therefore, the conversion from the discrete time domain to the continuous domain is a must when PEM is applied to a continuous system. The transformation used for this converting process could contribute to the noticeably larger bias appearing in these simulation studies. Another possible cause could be ascribed as the requisite steady state conditions and long data ensemble of the PEM was not met in our simulation studies.

2.5 Conclusion

The WLS/MFT and AWLS/MFT algorithms stemming from two different signal models have been devised and analyzed in detail. Under different assumptions, WLS/MFT is a maximum likelihood estimator and AWLS/MFT is an approximated maximum likelihood estimator. When the additive noise in the output is small, then the estimated parameters and covariance matrix from AWLS/MFT are fairly close to the results of the true maximum likelihood estimate. Lemma 4 has not only further disclosed the insightful relationships among the covariance matrices, but also paved the way for the numerical implementation of the WLS/MFT and AWLS/MFT algorithms. The recursive banded-sparse matrix inversion scheme in 2.3.3 provided a stabler and more efficient method of inverting the covariance matrices. From the simulation and comparison studies in section 2.4, the WLS/MFT and AWLS/MFT schemes have improved the previous LS/MFT method in both bias and variance, and both achieved a smaller variance than the popular PEM algorithm which has the worst bias results. Meanwhile, the simulations in 2.4 also show that the initial conditions have basically no visible affect on the performance of MFT algorithms, which is concordant with the theoretical analysis in section 2.1. Again, it has affirmed that the MFT method is a potent tool to cope with the identification problems using transient I/O data.

Chapter 3

Continuous Time Model Reduction Using AWLS/MFT Algorithm

3.1 Overview

Simplifying a high order or complex model with a lower order model has been deemed as one of the most important topics in automatic control, signal processing and other engineering and science areas. For many complicated high order models, the reduction not only can significantly facilitate their analysis and design, but also makes the digital or analog simulation and implementation possible and affordable. In the sense of approximation, the lower order models should be able to replicate the time domain,

e.g., impulse response or step response, as well as frequency response, e.g., Bode diagram, as closely to those of the high order model as possible. From the practical point of view, the following situations are very common in demanding simplified models:

1. Given a higher order system, ask reduction.

2. Given a complicated-looking Bode plot, request a simpler parametrized model.

3. Given an input/output data pair from an unknown-parameterized higher or nonlinear model, demand a model with a specified lower order.

During the last two decades, many research results on continuous time model reduction have been reported [25] [48] [50] [24] [14] [23]. Most of the earlier work, categorically named as classical reduction methods (CRM) [14], has been carried out based on classical mathematical approximation theories such as the Padè approximation [50] [48], the continued-fraction method, and the time-moment-matching method [57]. The essence of CRM schemes is expanding the original system into a Taylor series about the origin or the low frequency end, while neglecting the rest of the frequency range. This naturally incurs the possibility of low accuracy in the higher frequency band and potential loss of stability of the reduced system although the original system may have been stable. In order to obtain a stable model, many modified reduction schemes, like the stability-criterion and differentiation methods [23] can guarantee that the final low order model is stable by allotting some stable poles

to the denominator beforehand while letting the numerator be determined by the CRM. Unfortunately, in most cases, the accuracy in this kind of algorithm has fallen prey to the higher priority of stability. The FF-Padè (Frequency-fitting coupled with Padè approximation) method [14] has been proposed to alleviate these drawbacks by fitting the mid-frequency range.

Another class of schemes belonging to the time domain methods [25] computes the parameters of a reduced order model so as to minimize a certain criterion function characterized by the difference of time domain responses (typically impulse responses) to a given driving signal [25]. A reduction algorithm developed by Sakr and Bahgat [47] to obtain an optimal reduced order model for a power plant has been one of the examples of this kind of time domain approach.

Stemming from principal component analysis and singular value decomposition, Balanced-Realization [24] has proven to be a seasoned order reduction method in both theory and practice; it has been commercialized in the popular MATLAB control toolbox. The Balanced-Realization scheme was derived from the "signal injection" viewpoint by characterizing the relevant subspaces in terms of responses to injected signals. The model reduction is done by eliminating subsystems associated with small singular values.

One common ingredient of all the above schemes is that either high order transfer functions or the state space models must be known in advance in order to carry out the reduction. This means that they are only used to cope with Situation 1 listed

earlier.

The Modulating Function Technique (MFT) has been no stranger to this field. In [26], three nonparametric modulating function frequency matching (MFFM) schemes have been initiated and compared with other algorithms. MFFM is a two-step scheme: first estimate the frequency response of the system through an nonparametric modulating function algorithm [36] from several I/O data pairs; secondly, estimate the parameters of a reduced order model by minimizing a frequency matching criterion. Parametric LS/MFT has also been resorted to in chemical system reduction [6], though MFFM outperformed LS/MFT in [26]. With the more sophisticated AWLS/MFT algorithm presented here, better results are expected and will be demonstrated.

In this Chapter, the AWLS/MFT algorithm stated in the last chapter will be utilized to reduce the orders of higher order systems, and the comparison will be made with other published results and algorithms, especially with balanced realization, FF-Padè and MFFM schemes. As a final example, a 12th order power plant model given by Sakr and Bahgat [47] is reduced to a 2nd order model and compared with their time-domain-based reduction results. The versatility and flexibility of AWLS/MFT will be addressed in terms of handling the situations listed above. Some important key points of implementation will be mentioned as well.

3.2 AWLS/MFT and Order Reduction

AWLS/MFT has been developed to tackle continuous time parameter identification problems based on an I/O data pair. Parameters are obtained through minimizing a structure-specified time-domain differential equation model error modulated by a set of known modulating functions. Due to the Fourier type modulating functionals we have used, both time and frequency domain information has been naturally concatenated into the joint cost function (2.91). This viewpoint of AWLS/MFT is close to what model reduction is trying to accomplish. The essence of the two is so indistinguishable that AWLS/MFT should be able to shoulder the mission of model reduction. AWLS/MFT was developed as an I/O-data-pair-oriented scheme, which implies that for a given I/O pair of a high order or complex system (the exactly parametrized model of this system might not be available), AWLS/MFT can be applied to produce a model with a specified lower order. If the transfer function or state space representation of a high order system is given and a lower order model is desired, the I/O data pair can be acquired by driving the original system with a rich persistent signal, typically a Gaussian distributed random sequence, and simulating the output with the help of the LSIM routine of MATLAB, so that AWLS/MFT can then be used. Many mechanical systems or components are often characterized by weird-looking Bode diagrams attached to them, when they are sent out of the manufacturing or testing sites. If connecting these systems into control loops is desired, reduced parametrized models have to be acquired first. In this case, the driving signals

could be formed by frequencies coinciding with the modulating function frequencies which are decided upon from the Bode diagrams, while accordingly randomizing the corresponding phases. One judicious way of choosing the phases would be by using so-called low-noise noise [41] which is accomplished by picking phases minimizing the fourth moment of the driving sequence. This low-noise noise can smooth out the giant peaks which might otherwise drive the system into the nonlinear regions of system operation. (This might not be the case in model reduction.) The resulting low-noise signal makes the input look like random noise. Gaussian distributed randomized phases have been utilized in our simulation studies, and the results have been very satisfactory. Another major concern often encountered in model reduction is whether or not the reduced order model is stable. This concern can be easily eased with the stability-constrained AWLS/MFT algorithm described in the previous chapter which automatically locates all the poles of the resulting model in the stable region.

Technically, the following algorithm-related parameters must be specified before AWLS/MFT can be applied

Fs : Sampling Frequency
N : I/O Data Length (Number of samples)
ω_0 : Resolving Frequency $= 2\pi \cdot Fs/N$
ω_B : Modulating Bandwidth (System Bandwidth Covered by Modulating Frequencies)
M : Maximum Modulating Frequency Index = "integer-part (ω_B/ω_0)"

Among the above, ω_0 is the most intrinsic algorithm-related parameter which determines the frequency resolution, especially when accuracy is demanded in the low frequency range. In some cases in which high accuracy is desired in the middle or high frequency range, ω_o can be set relatively larger so as to alleviate the computation

burden while preserving good accuracy. This flexibility is exclusively possessed by MFT-based schemes. From the several examples we are going to present here, the roles of these parameters will be further illuminated.

3.3 Comparison with Other Methods

In order to provide some quantitative measures of goodness in replicating the original systems characteristics, the signal-to-error ratios (SER) or S/E are defined, in both time and frequency domains respectively, as

$$\text{SER}_t = 20 \cdot \log_{10}\left\{\frac{\|h_o(t)\|_2}{\|h_o(t) - h_r(t)\|_2}\right\}, \qquad \text{in time domain } t \in [0, T_s] \tag{3.1}$$

$$\text{SER}_f = 20 \cdot \log_{10}\left\{\frac{\|H_o(j\omega)\|_2}{\|H_o(j\omega) - H_r(j\omega)\|_2}\right\}, \qquad \text{in frequency domain } \omega \in [\omega_1, \omega_2] \tag{3.2}$$

where $\|\cdot\|_2$ denotes the L_2 norm in the appropriate space, and

$h_o(t)$	:	time response (e.g., step or impulse response) of original system.
$h_r(t)$	:	time response (e.g., step or impulse response) of reduced system.
$H_o(j\omega)$	:	frequency response of original system.
$H_r(j\omega)$	:	frequency response of reduced system.
T_s	:	time interval of interest (roughly the settling time of the system).
ω_1 and ω_2	:	frequency range of interest.

Without specific mentioning, all the dB numbers in the graphs of this chapter should mean SER or S/E values[1].

[1] In all six examples, the SER_f numbers will be calculated at frequency nodes generated by the MATLAB routine LOGSPACE($\omega_1, \omega_2, 100$), where $[\omega_1, \omega_2]$ is the same as the graphic range appearing in each magnitude or phase plot. Therefore, for the high frequency matching in Example 1 and 2, those SER numbers are less indicative due to less concern about the low frequency range.

Example 1: As the first example of our comparison studies, a sixth order low-pass system:

$$H_1(s) = \frac{(s+2)^2 \cdot (s+5)^2 \cdot (s+100)}{(s+1)^2 \cdot (s+10)^2 \cdot (s+100)^2} \tag{3.3}$$

from [26] is chosen to be reduced to a 3rd order model. Using this system, Co and Ydsti [6] compared the FF-Padè algorithm with LS/MFT scheme and found that LS/MFT worked better than the FF-Padè method. In Pan's thesis [26], MFFM gave a better fit than the LS/MFT scheme with the following low frequency matching model:

$$\tilde{H}_1^{MFFM} = \frac{4.5233s^2 + 30.6739s + 45.2871}{s^3 + 51.2456s^2 + 710.2395s + 453.6771} \tag{3.4}$$

which was obtained through eight-seconds of data at a sampling rate $F_s = 200$Hz, i.e., 1600 simulation points. From MATLAB, the reduced 3rd order model using Balanced-Realization routine MODRED has been acquired as

$$H_1^{B-R}(s) = \frac{5.4156s^2 + 30.9466s + 58.5745}{s^3 + 65.4461s^2 + 793.1840s + 585.7454} \tag{3.5}$$

Choosing $F_s = 64$(Hz), $\omega_B = 10\pi$(rad/s) and $N = 1024$ (resolving frequency: $\omega_0 = 0.125\pi$(rad/s)[2]) and using a Gaussian random noise sequence to excite the $H_1(s)$,

[2]In this case, the curvature of the low frequency band is slowly-changing, so that $\omega_0 = 0.125\pi$(rad/s) should be fine enough. If ω_0 is further reduced by raising N, it will not make any significant difference. Decreasing ω_0 by lowering the F_s is not recommended by and large, due to its potential influence on the accuracy in numerical integration. Please refer to Examples 3~6 in this Chapter for the cases in which a much finer resolving frequency is required to identify sharply-changing peaks and valleys in the Bode diagrams.

AWLS/MFT gives the low frequency matching model:

$$\tilde{H}_1^{AWLS} = \frac{4.7801s^2 + 28.9623s + 50.4280}{s^3 + 57.8232s^2 + 713.7344s + 502.0671} \tag{3.6}$$

The Magnitude, phase and step response plots of the models from Balanced-Realization, MFFM and AWLS/MFT algorithms are summarized in Figures 3.1(a), 3.1(b), 3.2 and 3.3 respectively. In the frequency domain, the three algorithms have very

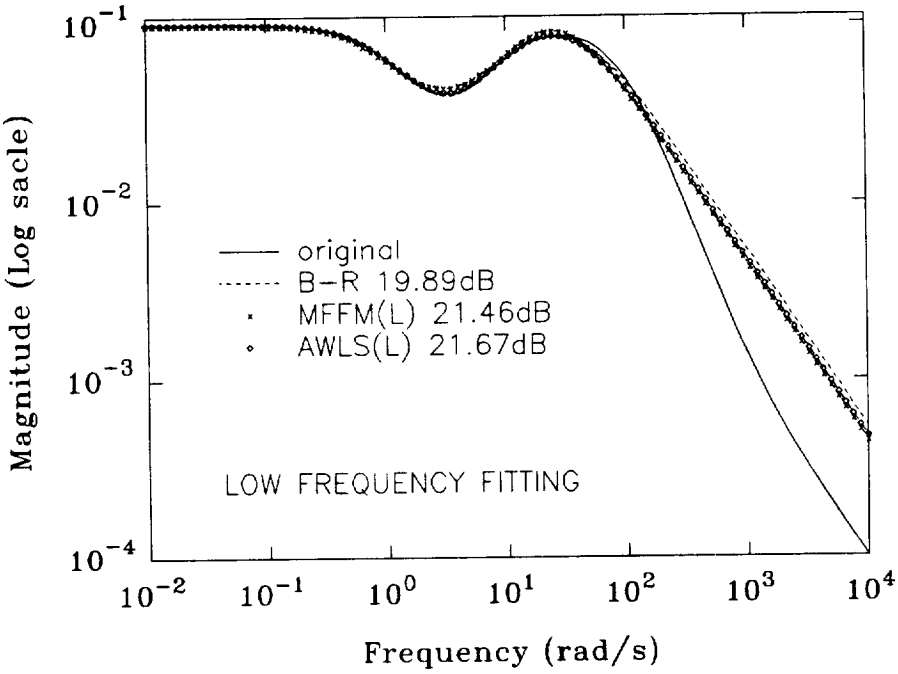

Figure 3.1(a): Log-scale magnitude plots of the low frequency matching models.

close SER numbers, though the AWLS/MFT has a slight numerical edge. Graphically, especially from the linear-scale magnitude plot and phase plot, the AWLS fits best in the low frequency side (about a 2dB lead). From the step response plots, Balanced-Realization and AWLS/MFT are better than MFFM while the visual difference between Balanced-Realization and AWLS/MFT is negligible. Numerically,

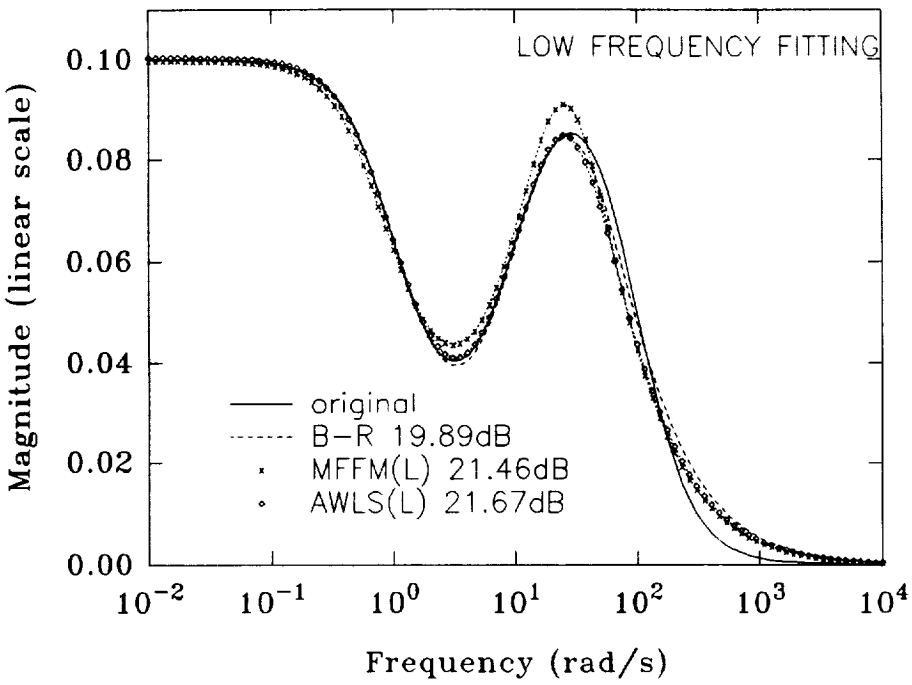

Figure 3.1(b): Linear-scale magnitude plots of the low frequency matching models.

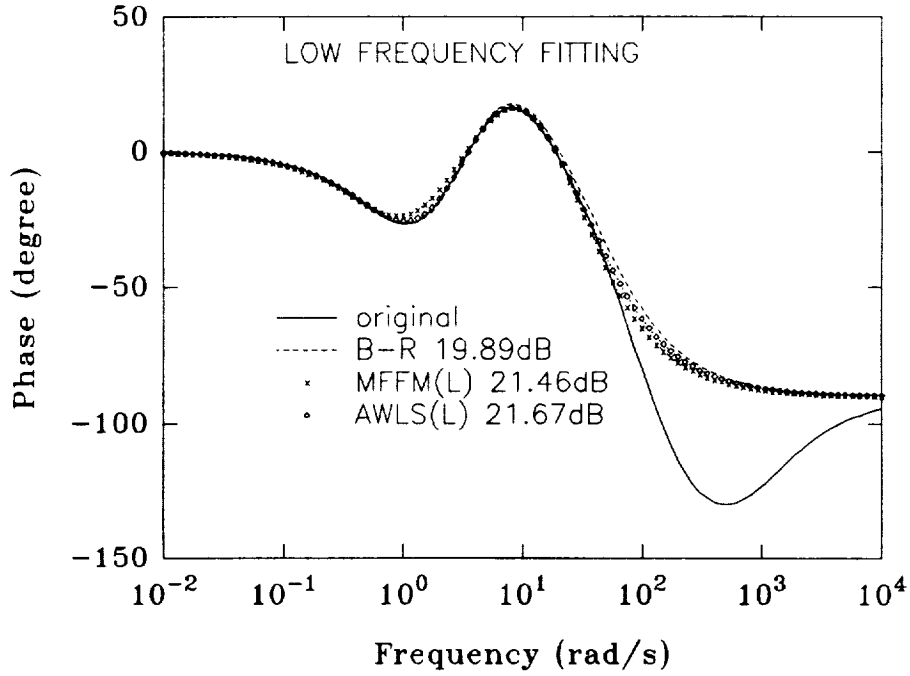

Figure 3.2: Phase plots of the low frequency matching models.

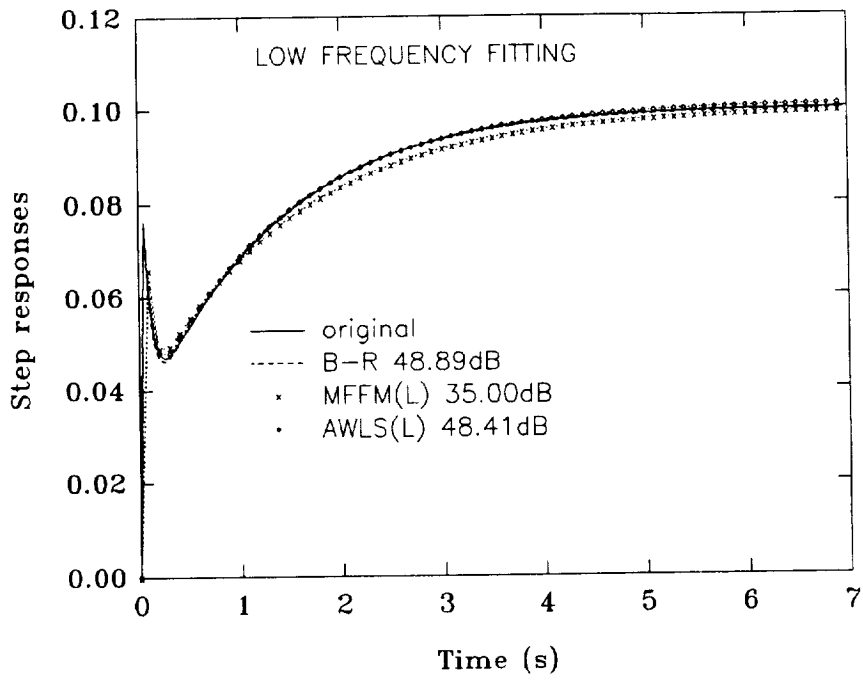

Figure 3.3: Step responses of the models.

the Balanced-Realization has a very small lead (about 0.5dB).

As mentioned before, MFFT and AWLS are capable of reducing the original system to a frequency-range-specified low order model through freely choosing the frequency range to match. Pan [26], with this very same example, manifested this kind of flexibility and gave a high frequency matching model:

$$H_1^{MFFM}(s) = \frac{1.4952s^2 + 863.1844s + 5334.4197}{s^3 + 187.4456s^2 + 11304.2301s + 141250.0129} \tag{3.7}$$

With $F_s = 1024$(Hz), $\omega_B = 100\pi$(rad/s), $N = 256$ (resolving frequency: $\omega_0 = 8\pi$(rad/s)[3]) and a Gaussian random noise sequence exciting the $H_1(s)$, AWLS/MFT

[3]Like the low frequency matching, the high frequency band also looks "smooth" in this case, and the value $\omega_0 = 8\pi$(rad/s) is fine enough. By changing N to make ω_0 relatively coarser or finer, it

came up the high frequency (roughly from $8\pi \sim 100\pi$(rad/s)) matching model:

$$H_1^{AWLS}(s) = \frac{1.010s^2 + 1003.8802s + 6415.0895}{s^3 + 214.2575s^2 + 12816.7076s + 148767.3293} \tag{3.8}$$

In order to see the high frequency matching of the MFFM and AWLS schemes clearly, the frequency responses of the two reduced models are drawn together in Figures 3.4(a), 3.4(b) and 3.5. Clearly, AWLS/MFT is favored both graph-

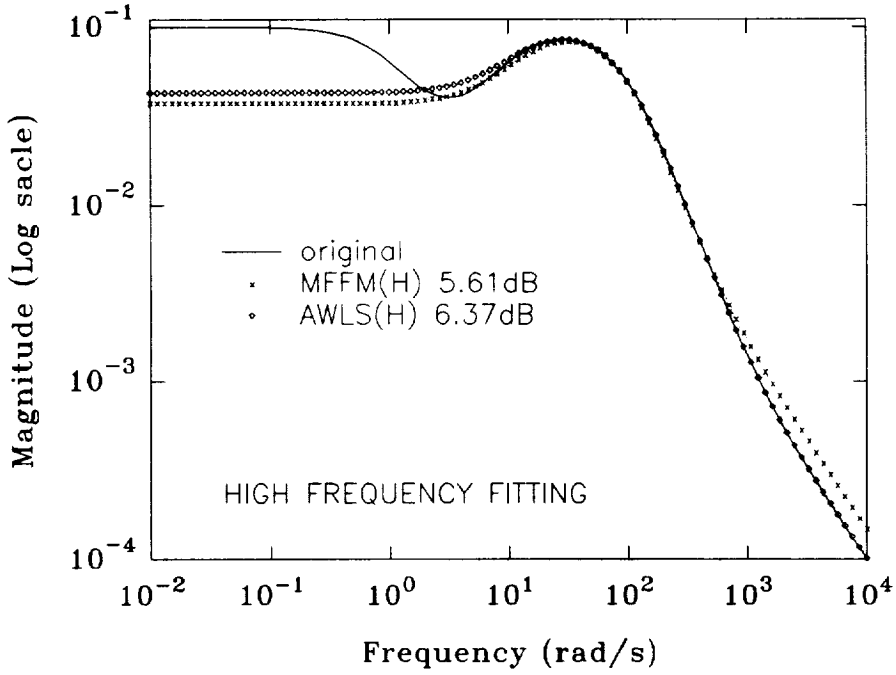

Figure 3.4(a): Log-scale magnitude plots of the high frequency matching models.

ically and numerically. Another aspect we should notice is that AWLS/MFT used only 256 points, equivalent to 250ms of I/O data, and took practically no time to get its model, while MFFM used 1600 points which is eight-seconds of data. Therefore,

will not cause any noticeable change. However, if there is a kink in the high frequency range like Example 2 of this Chapter, a finer ω_0 is a requisite.

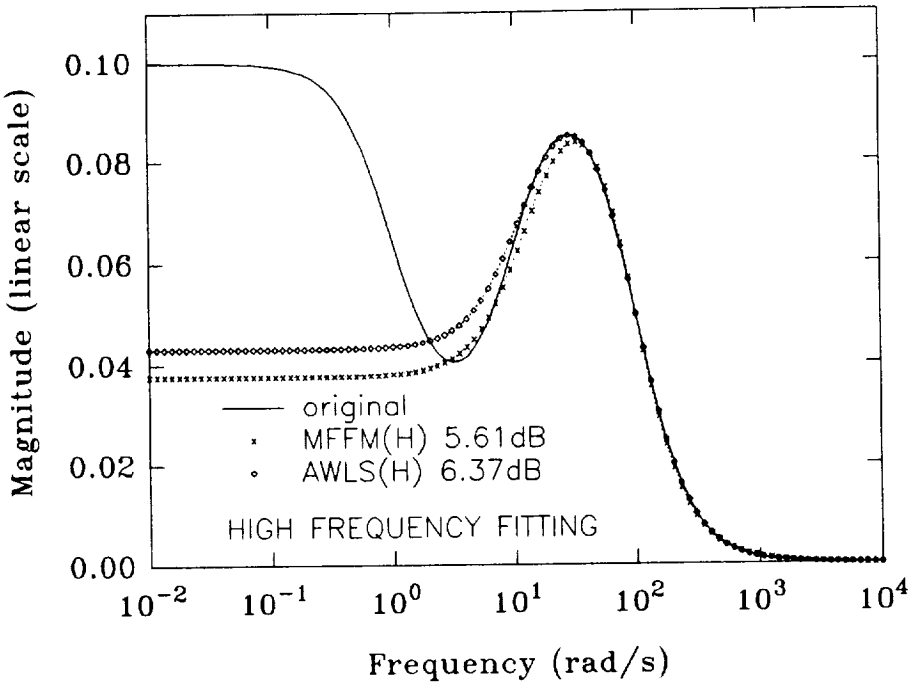

Figure 3.4(b): Linear-scale magnitude plots of the high frequency matching models.

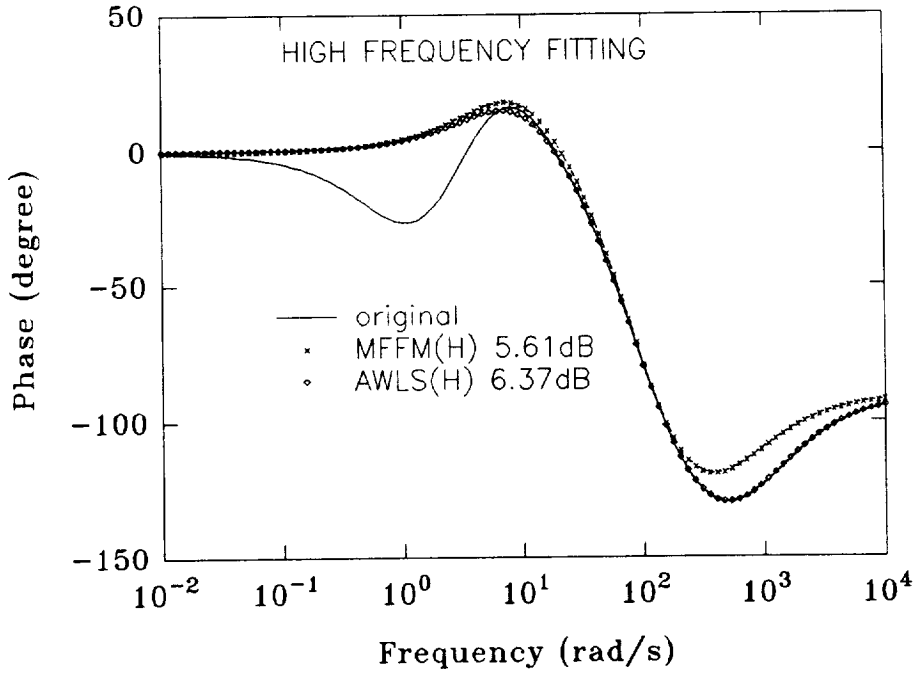

Figure 3.5: Phase plots of the high frequency matching models.

AWLS/MFT can do high frequency fitting more efficiently.

Example 2: We now consider the following sixth order model [26]:

$$H_2(s) = \frac{4.5s^5+16.8750s^4+1.2474\times10^4s^3+5.3034\times10^4s^2+8.0454\times10^6s+3.6556\times10^7}{s^6+66.85s^5+2778.9s^4+7.1963\times10^4s^3+1.0168\times10^6s^2+9.8011\times10^6s+4.7306\times10^7} \tag{3.9}$$

Pan again used eight-seconds of data and 1600 data points and by weighting the middle frequency range slightly more, came up with the following reduced third order model:

$$H_2^{MFFM}(s) = \frac{3.8121s^2 - 59.6845s + 6913.6012}{s^3 + 42.7534s^2 + 389.3838s + 8388.2031} \tag{3.10}$$

This model is basically obtained through a high frequency matching, while he did not provide a low frequency match model in this example. From MATLAB, the reduced model using Balance-Realization is given by

$$H_2^{B-R}(s) = \frac{-5.5419s^2 + 170.4683s + 872.5521}{s^3 + 15.4064s^2 + 207.4581s + 1129.1082} \tag{3.11}$$

Like the last example, we can still exploit the flexibility of MFT algorithms by fitting different frequency ranges. For a low frequency range, setting $F_s = 256(\text{Hz})$, $\omega_B = 2\pi(\text{rad/s})$ and $N = 1024$ (resolving frequency:$\omega_0 = 0.5\pi(\text{rad/s})$), we have

$$\tilde{H}_2^{AWLS}(s) = \frac{-4.521s^2 + 155.179s + 704.284}{s^3 + 14.216s^2 + 196.103s + 914.840} \tag{3.12}$$

If we change the setting to $F_s = 1024(\text{Hz})$, $\omega_B = 50\pi(\text{rad/s})$ and $N = 1024$ (resolving frequency:$\omega_0 = 2\pi(\text{rad/s})$), the high frequency matching model is obtained as

$$H_2^{AWLS}(s) = \frac{4.5342s^2 + 5.1391s + 6581.3344}{s^3 + 45.0462s^2 + 385.1122s + 6394.8751} \tag{3.13}$$

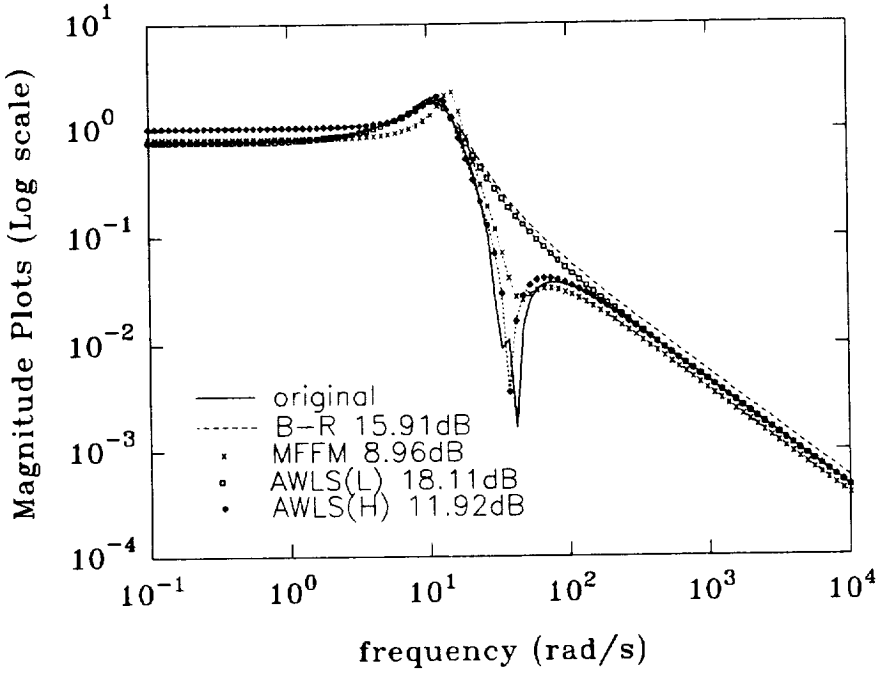

Figure 3.6(a): Log-scale magnitude plots of the models.

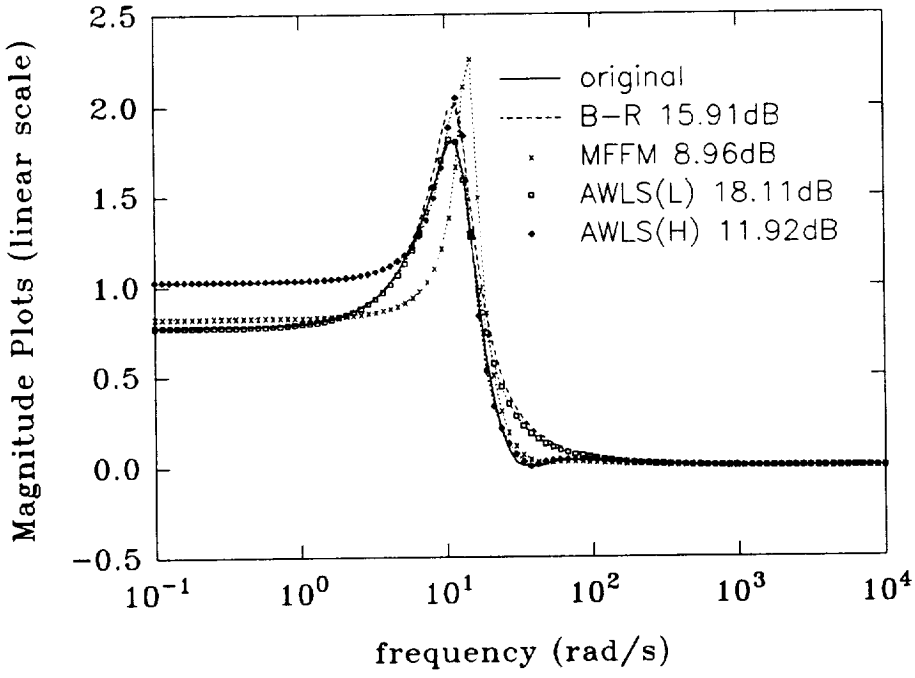

Figure 3.6(b): Linear-scale magnitude plots of the models.

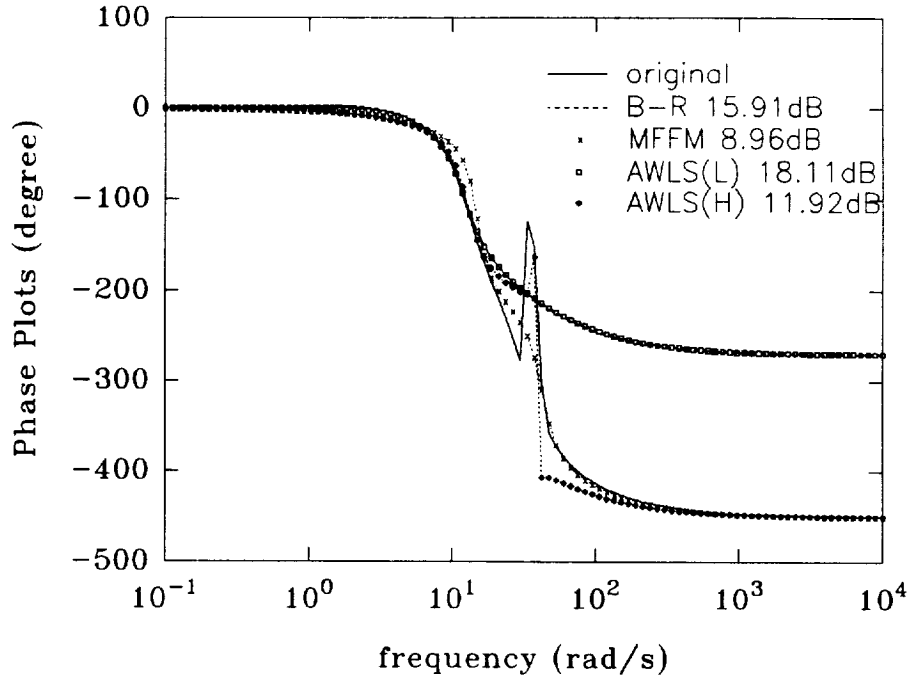

Figure 3.7: Phase plots of the models.

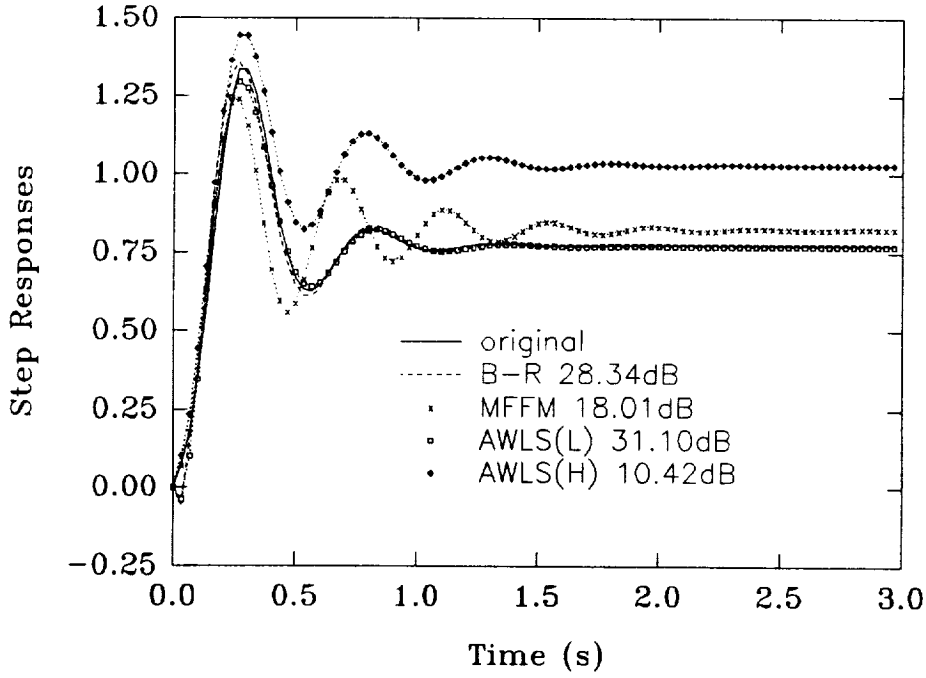

Figure 3.8: Step responses of the models.

The performance of these models can be evaluated from Figure 3.6(a), 3.6(b), 3.7 and 3.8. For the low frequency range in the frequency domain, AWLS/MFT leads balanced-realization numerically (18.11/15.91) and the only difference that can be observed is that in Figure 3.6(b) Balanced-Realization is off around the peak region, while AWLS/MFT coheres to that kink. For the high frequency matching, AWLS/MFT does not have a biased peak like MFFM shown in Figure 3.6(b), while a magnitude deviation from the original system in the very high frequency range appeared in Figure 3.6(a). As for that notch concern, AWLS(H)/MFT is much closer than the other, as seen in Figure 3.6(a); numerically, AWLS/MFT is better as well. In the step response plots, the $\tilde{H}_2^{AWLS}(s)$ has the best numerical result, followed very closely by the Balanced-Realization scheme. It is no surprise that both of the high frequency matching models, $H_2^{AWLS}(s)$ and $H_2^{MFFM}(s)$, are not even close to the true step response, because after the initial shoot-up the step response is primarily determined by low frequency characteristics of the system. Again, AWLS/MFT took a much shorter length of data in acquiring the above high frequency matching model.

Example 3: The following sixth order model was the second example used in [14] to compare with other classical Padè approximation methods:

$$H_3(s) = \frac{(1+2.0587s)(1+2.5529s+5.4342s^2)(1+3.2648s+2.1476s^2)}{(1+3.0092s+0.7970s^2)(1+6.8538s+0.6965s^2)(1+0.1394s+0.6861s^2)} \tag{3.14}$$

The FF-Padè scheme reduced the above sixth order system into the following third

order model:

$$H_3^{FF-Pade}(s) = \frac{1 - 1.4257s + 4.3109s^2}{1 + 0.7003s + 0.8613s^2 + 0.0837s^3} \tag{3.15}$$

The MATLAB routine of the Balanced-Realization scheme gave the reduced model as:

$$H_3^{B-R}(s) = \frac{110.8197s^2 + 26.4163s + 38.2340}{s^3 + 26.7135s^2 + 7.4733s + 38.2340} \tag{3.16}$$

With $F_s = 256(\text{Hz})$, $\omega_B = 5\pi(\text{rad/s})$ and $N = 8192$ (resolving frequency: $\omega_0 = 0.0625\pi(\text{rad/s})$), AWLS/MFT produced the third order model:

$$H_3^{AWLS}(s) = \frac{67.3600s^2 + 6.4604s + 18.8969}{s^3 + 13.7774s^2 + 4.8674s + 18.8969} \tag{3.17}$$

The frequency and time responses are plotted in Figures 3.9(a), 3.9(b), 3.10 and 3.11.

The AWLS/MFT and Balanced-Realization performed reasonably well, though AWLS has shown a slight edge in the frequency domain while Balanced-Realization has led in the time domain. The FF-Padè lags far behind numerically and graphically. Frankly to say, this is the toughest example for AWLS/MFT we have met in all our model reduction studies in the sense of data length required for a very fine resolving frequency to identify that narrow valley-peak transition band.

Example 4: Another middle and high frequency range model used in [14] is the sixth order high-pass system:

$$H_4(s) = \frac{1 + 8.8818s + 29.9339s^2 + 67.087s^3 + 80.3787s^4 + 68.6131s^5}{1 + 7.6194s + 21.7611s^2 + 28.4472s^3 + 16.5609s^4 + 3.5338s^5 + 0.0462s^6}. \tag{3.18}$$

Using the middle range frequency fitting and Padè approximation for the lower fre-

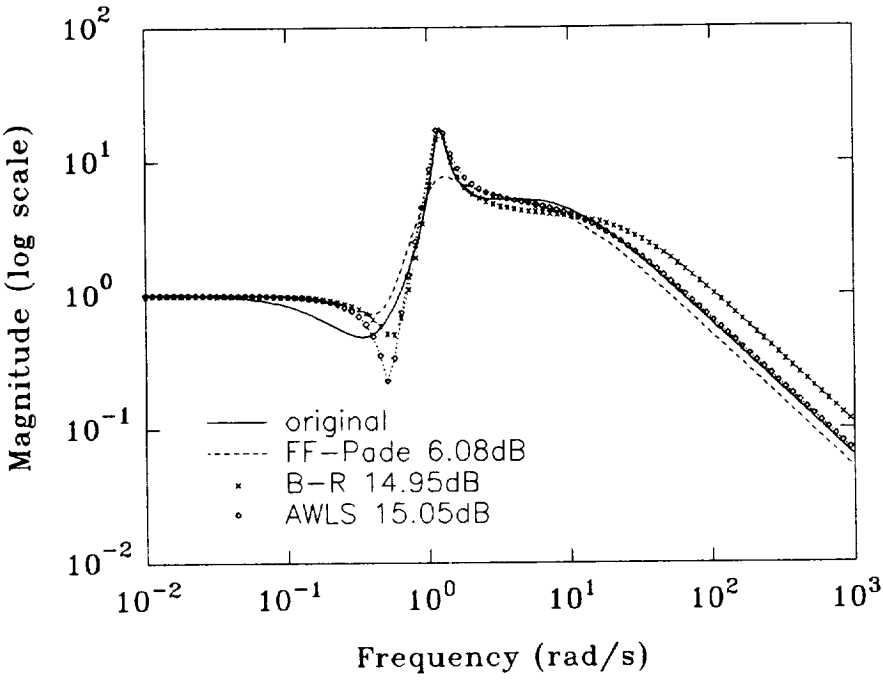

Figure 3.9(a): Log-scale magnitude plots of the models.

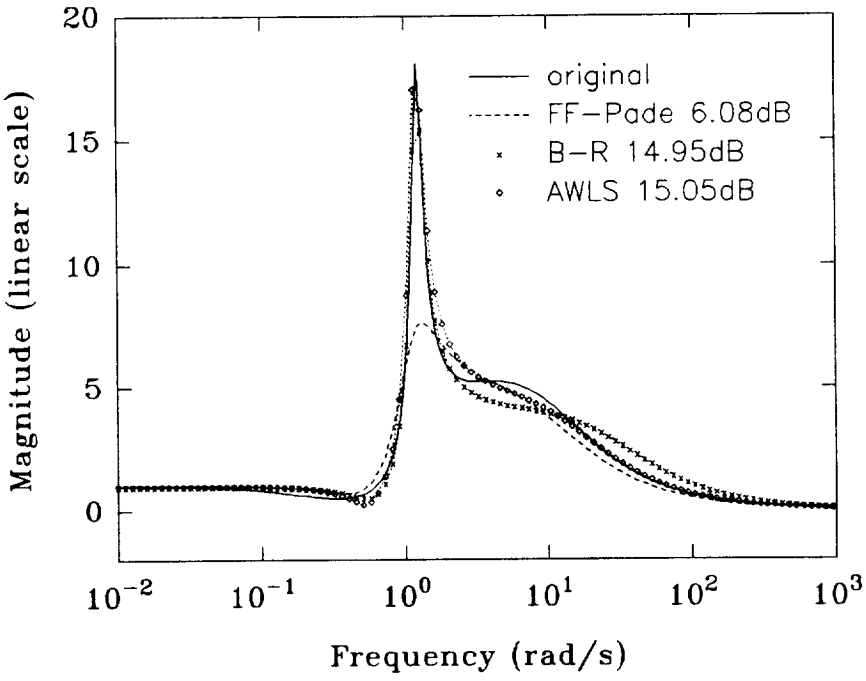

Figure 3.9(b): Linear-scale magnitude plots of the models.

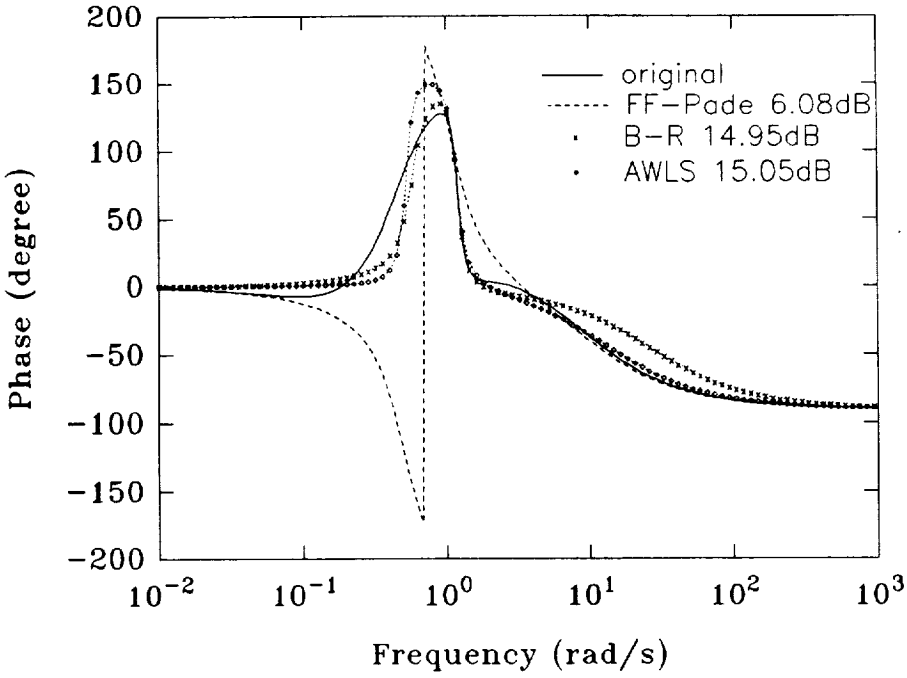

Figure 3.10: Phase plots of the models.

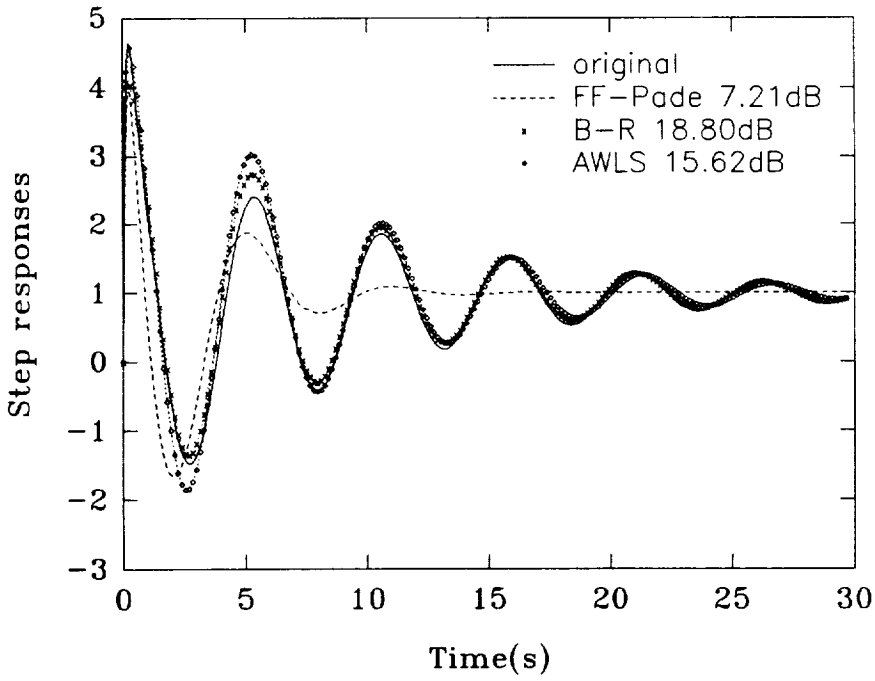

Figure 3.11: Step responses of the models.

quency band, FF-Padè had the following reduced third order model:

$$H_4^{FF-Pade}(s) = \frac{1 + 2.0098s + 3.7169s^2}{1 + 0.7474s + 0.1898s^2 + 2.4977s^3} \tag{3.19}$$

It should be noted that this reduced model $H_4^{FF-Pade}(s)$ is an unstable system, so that it would be unfair to compare its time domain response with the others. The Balanced-Realization model reduction scheme laid out its own reduced model as

$$H_4^{B-R}(s) = \frac{1551.8019s^2 + 554.9813s + 390.9247}{s^3 + 80.3317s^2 + 299.7585s + 390.9247} \tag{3.20}$$

Armed with the $F_s = 128(\text{Hz})$, $\omega_B = 2\pi(\text{rad/s})$ and $N = 8192$ (resolving frequency: $\omega_0 = 0.0312\pi(\text{rad/s})$), AWLS/MFT has reduced the $H_4(s)$ to

$$H_4^{AWLS}(s) = \frac{1466.9023s^2 + 342.8845s + 355.4733}{s^3 + 75.2876s^2 + 281.7353s + 319.1008}, \tag{3.21}$$

which is a stable system. The frequency domain comparisons are shown in Figures 3.12(a), 3.12(b), 3.13. The step responses of the Balanced-Realization and AWLS/MFT systems are plotted in Figure 3.14.

In comparing the frequency domain results for AWLS/MFT and Balanced-Realization algorithms, FF-Padè is absolutely in no sense a comparable method. Due to the slightly better fitting of AWLS/MFT around the pass-band peak area appearing in Figure 3.12(b), AWLS/MFT has a higher SER value. In the time domain response of Figure 3.14, both methods give almost a perfect match, while AWLS/MFT has a negligibly small numerical lead.

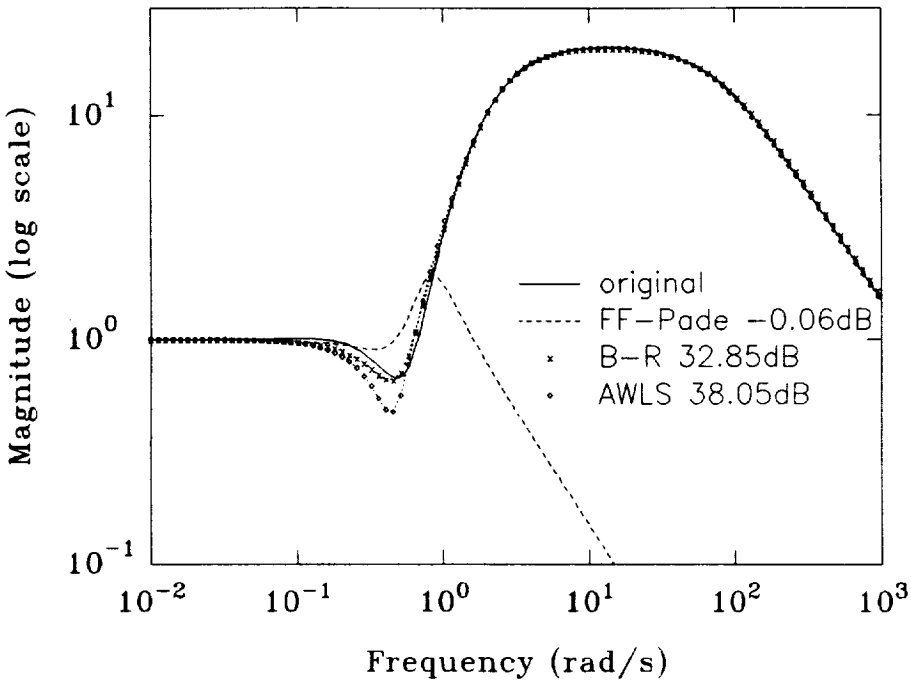

Figure 3.12(a): Log-scale magnitude plots of the models.

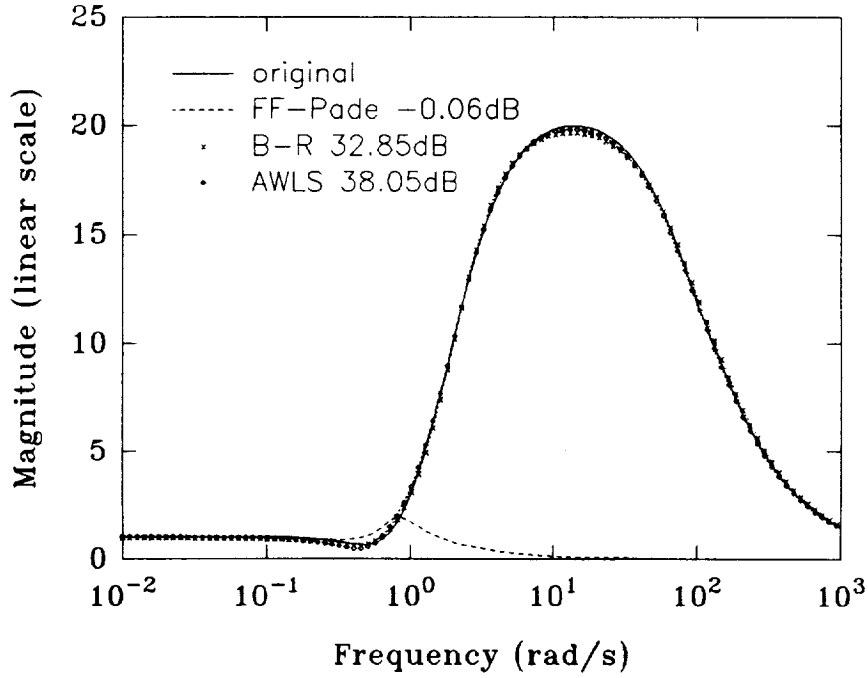

Figure 3.12(b): Linear-scale magnitude plots of the models.

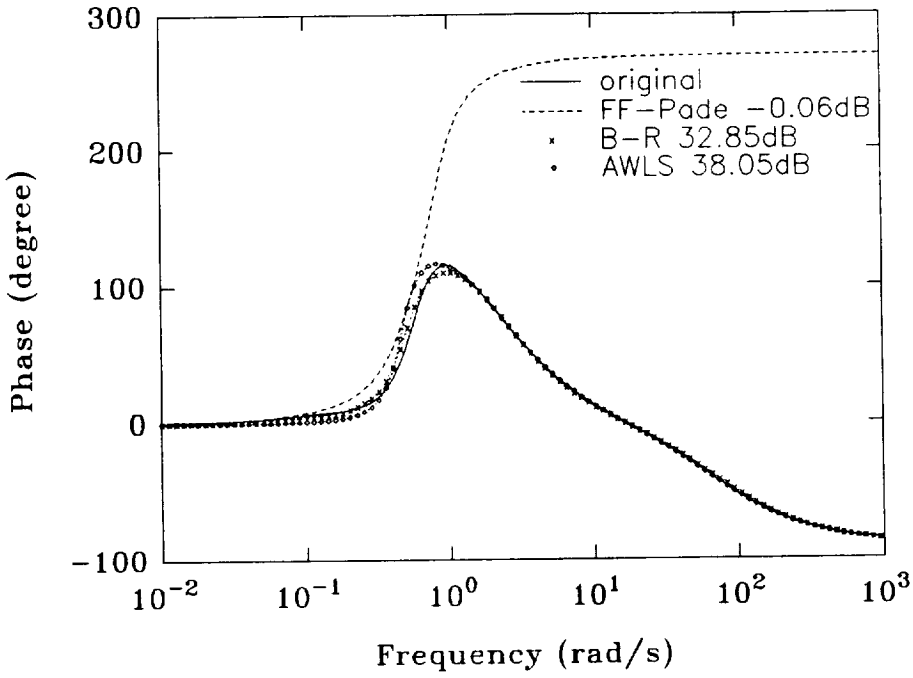

Figure 3.13: Phase plots of the models.

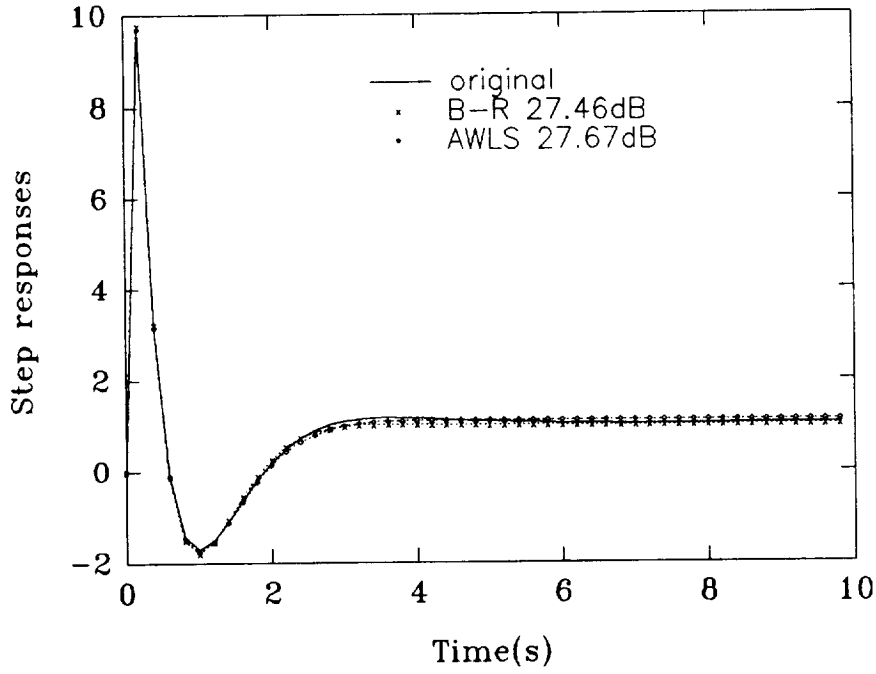

Figure 3.14: Step responses of the models.

Example 5: The last example used in [14] is a complex sixth order system:

$$H_5(s) = \frac{1 + 7.7617s + 13.5756s^2 + 67.6016s^3 + 40.2492s^4 + 144.0994s^5}{1 + 14.8243s + 75.7619s^2 + 163.2959s^3 + 139.3768s^4 + 38.6263s^5 + 3.3282s^6}. \tag{3.22}$$

The FF-Padè method gave the reduced fourth order model:

$$H_5^{FF-Pade}(s) = \frac{1 + 1.1483s + 4.5589s^2 + 5.0011s^3}{1 + 8.2109s + 5.2508s^2 + 1.4141s^3 + 0.200s^4}. \tag{3.23}$$

Again from MATLAB, the Balanced-Realization scheme gave

$$H_5^{B-R}(s) = \frac{50.9400s^3 + 68.1574s^2 - 8.3170s + 116.8597}{s^4 + 15.9582s^3 + 56.3846s^2 + 114.1988s + 16.8599}. \tag{3.24}$$

Running AWLS/MFT with $F_s = 32$(Hz), $\omega_B = 2\pi$(rad/s) and $N = 4096$ (resolving frequency: $\omega_0 = 0.0156\pi$(rad/s)), the following reduced fourth order system has been reached:

$$H_5^{AWLS}(s) = \frac{43.3696s^3 + 3.2472s^2 + 9.1180s + 3.3524}{s^4 + 11.5040s^3 + 39.1397s^2 + 39.8916s + 3.1557} \tag{3.25}$$

The frequency and time responses of all models are presented in Figures 3.15(a), 3.15(b), 3.16 and 3.17 respectively. In this example, even though FF-Padè has a relatively good fit in the notch part of Figure 3.15(a), the overall AWLS/MFT and Balanced-Realization are still better as shown in Figure 3.15(b) and 3.16, while AWLS/MFT has the closest frequency fitting, especially in the peak area, numerically and graphically. In the time domain, except for the large over-shoot of FF-Padè at an early stage, they all agree well with the true step response, though the Balanced-Realization has a slight lead over AWLS/MFT numerically.

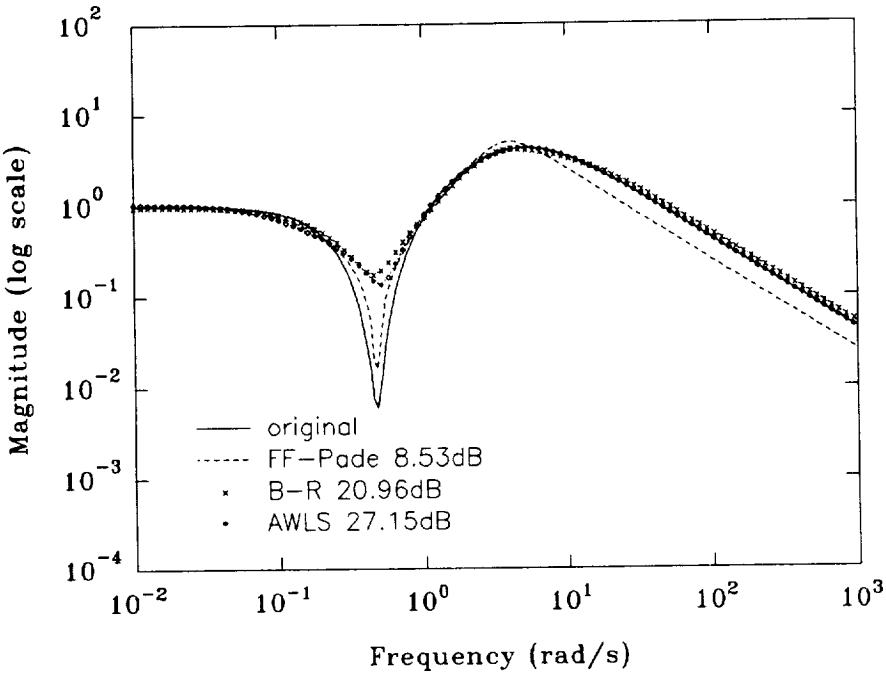

Figure 3.15(a): Log-scale magnitude plots of the models.

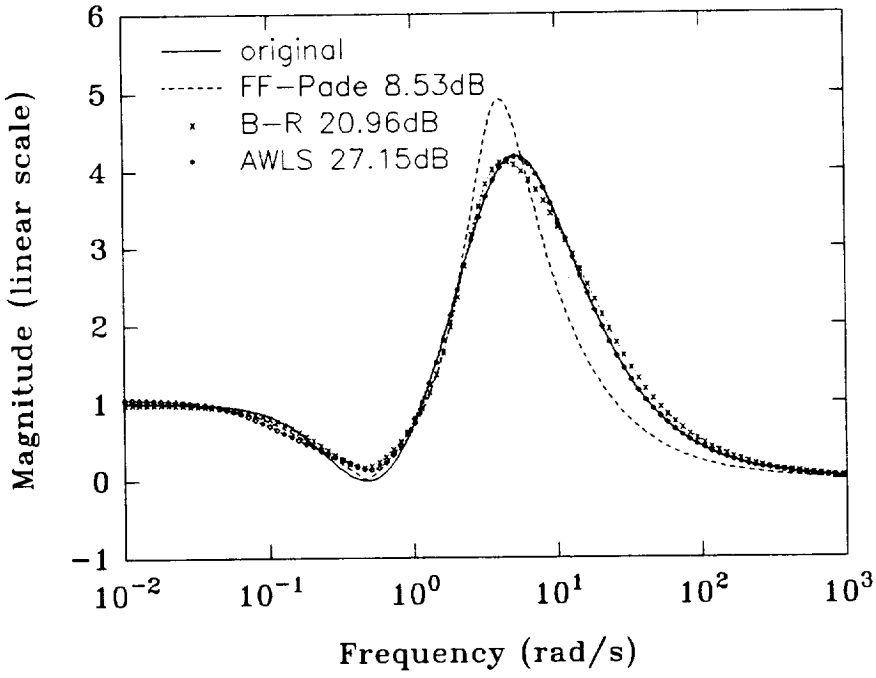

Figure 3.15(b): Linear-scale magnitude plots of the models.

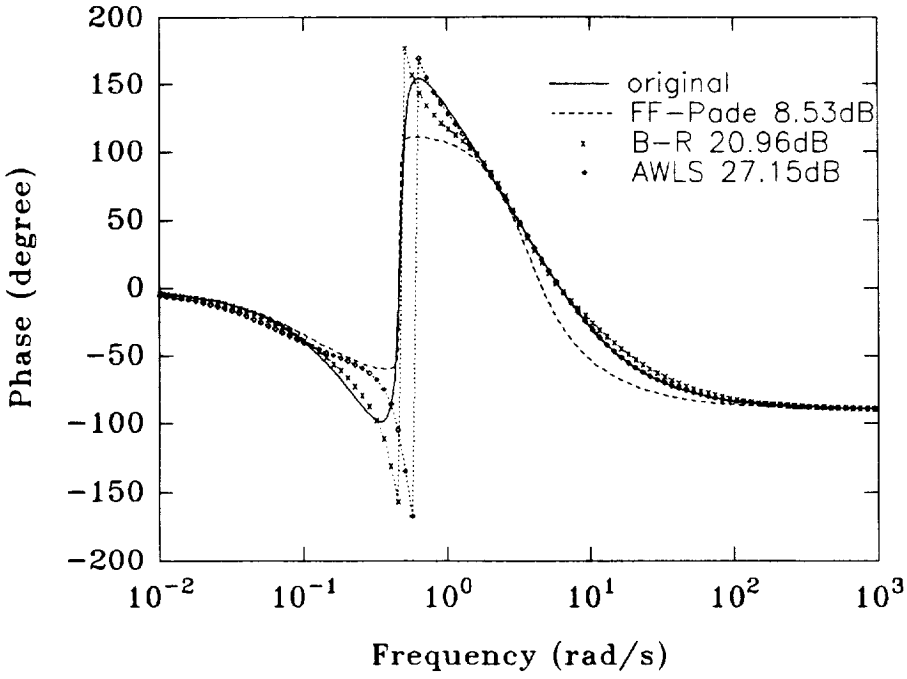

Figure 3.16: Phase plots of the models.

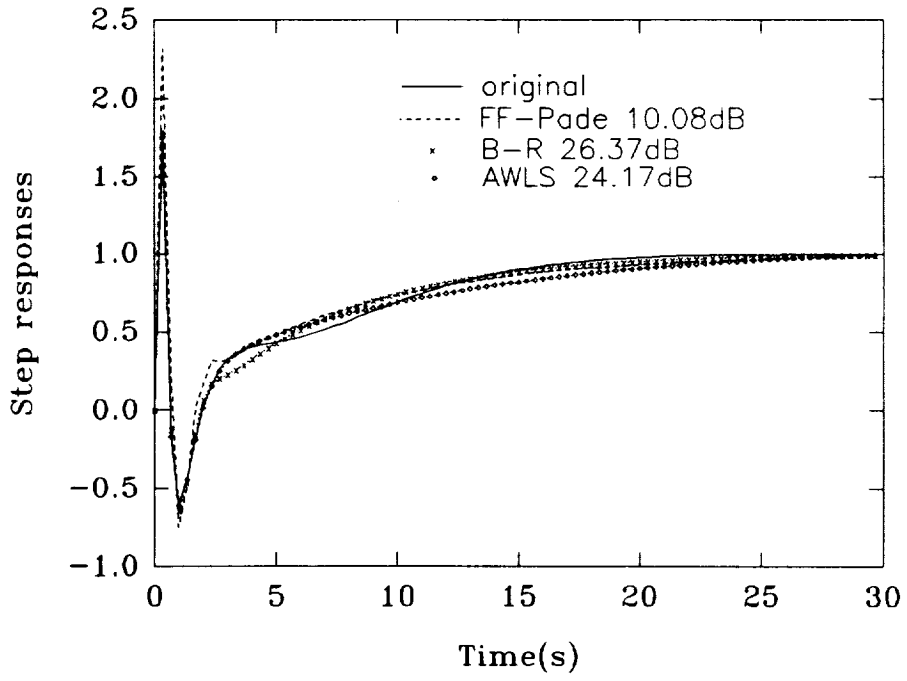

Figure 3.17: Step responses of the models.

Example 6: As the final example of our model reduction studies, we shall reduce a twelfth order power plant system arisen in a power generating station [47] to a non-strictly-proper second order model and compare it with the results stated in [47]. Sakr and Bahgat gave the following transfer function bridging between the output power and the input disturbance:

$$H_6(s) = \frac{684s^{11} - 93484.9s^{10} - 1.76 \times 10^6 s^9 - 3.25 \times 10^7 s^8 - 3.336 \times 10^8 s^7}{s^{12} + 16.9s^{11} + 335.88s^{10} + 3694.9s^9 + 30709s^8 + 199002s^7 + 928769s^6}$$

$$\frac{-1.944 \times 10^9 s^6 - 7.079 \times 10^9 s^5 - 1.697 \times 10^{10} s^4 - 2.71 \times 10^{10} s^3}{+3.02 \times 10^6 s^5 + 6.81 \times 10^6 s^4 + 1.047 \times 10^7 s^3 + 1.056 \times 10^7 s^2}$$

$$\frac{-2.80 \times 10^{10} s^2 - 1.703 \times 10^{10} s - 4.926 \times 10^9}{+6.344 \times 10^6 s + 1.7121 \times 10^6}. \tag{3.26}$$

Based on minimizing the cost function constructed with the time domain responses, the parameters of a non-strictly proper 2nd order system is optimally computed and this method will be denoted as the S-B scheme in our following discussion. In [47], the S-B algorithm with steady state constraint seems to give much better results than the one without the steady state constraint, so that only the reduced model with the steady constraint will be compared with the AWLS/MFT here. With the steady state constraint, S-B has reduced $H_6(s)$ to

$$H_6^{S-B}(s) = \frac{6365.3s - 142182.23}{s^2 + 1.475s + 52.311} + 2718. \tag{3.27}$$

Using BALREAL and MODRED routines of MATLAB, Balanced-Realization gives

$$H_6^{B-R}(s) = \frac{-266.884s^2 + 1112.084s - 142495.032}{s^2 + 0.5246s + 49.5261}. \tag{3.28}$$

Setting $N = 2048, \omega_B = 2\pi$ and $Fs = 50$Hz (resolving frequency: $\omega_0 = 0.048\pi$(rad/s)), AWLS/MFT has the reduced model:

$$H_6^{AWLS}(s) = \frac{68.477s^2 + 532.948s - 125098.5}{s^2 + 0.5235s + 49.3780}. \tag{3.29}$$

The frequency and time responses of the reduced models are plotted in Figures 3.18(a), 3.18(b), 3.19 and 3.20.

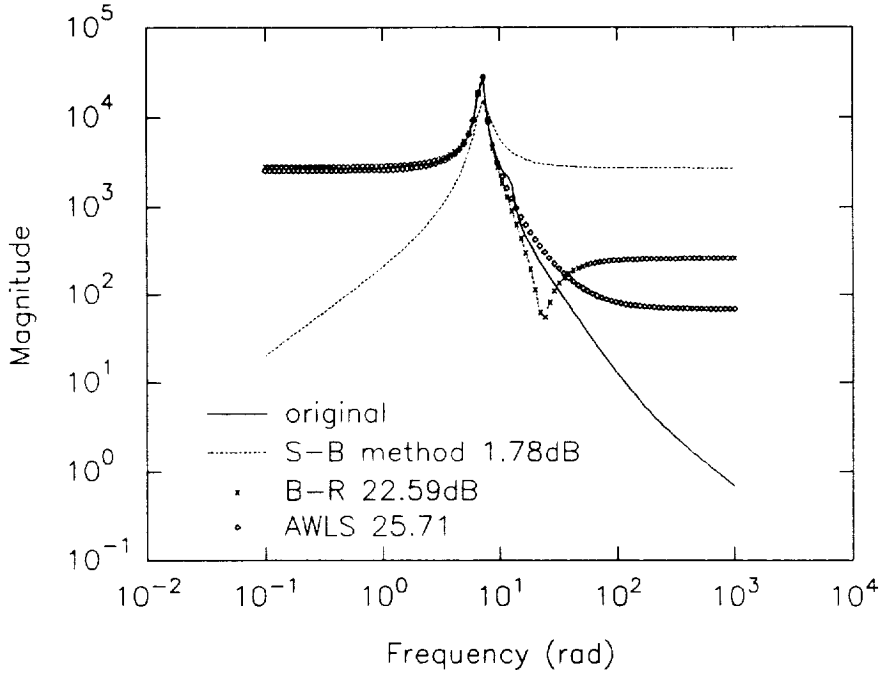

Figure 3.18(a): Log-scale magnitude plots of the models.

In the time domain, Figure 3.20, where S-B has been the derived, the AWLS/MFT and Balanced-Realization have accurately located the impulse response of the original system, while the impulse response from the S-B model is dying out too fast and its discrepancy from the true system is obvious, both visually and numerically. From the frequency response plots, Figures 3.18(a), 3.18(b) and 3.19, the AWLS/MFT and

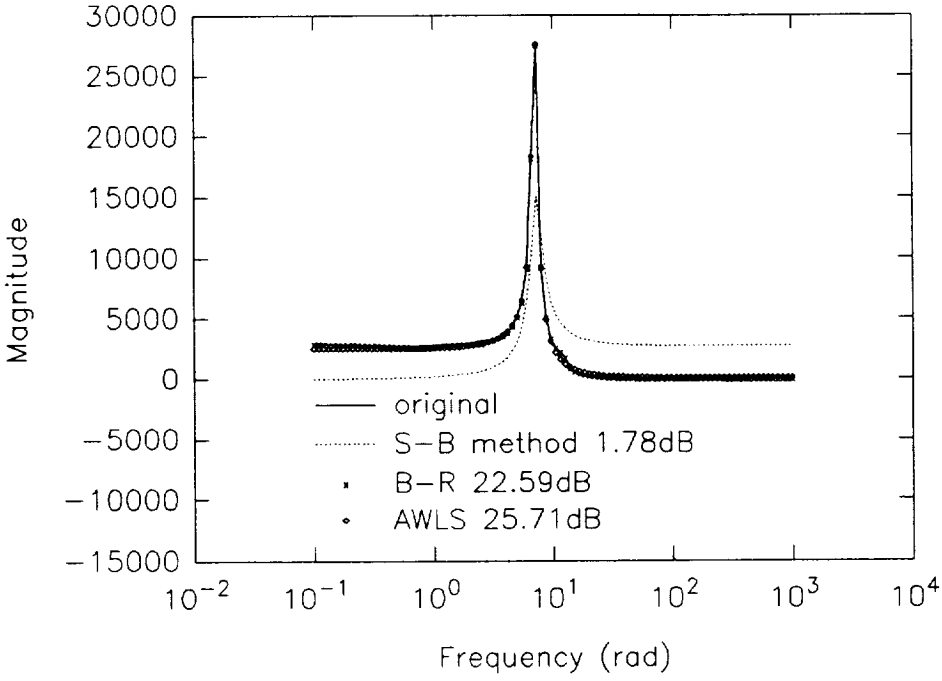

Figure 3.18(b): Linear-scale magnitude plots of the models.

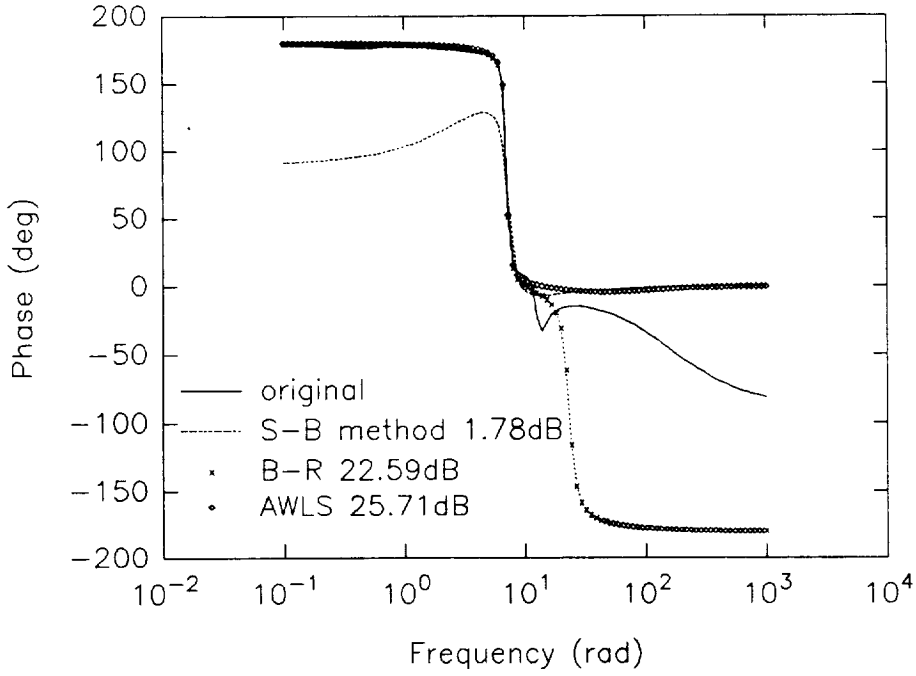

Figure 3.19: Phase plots of the models.

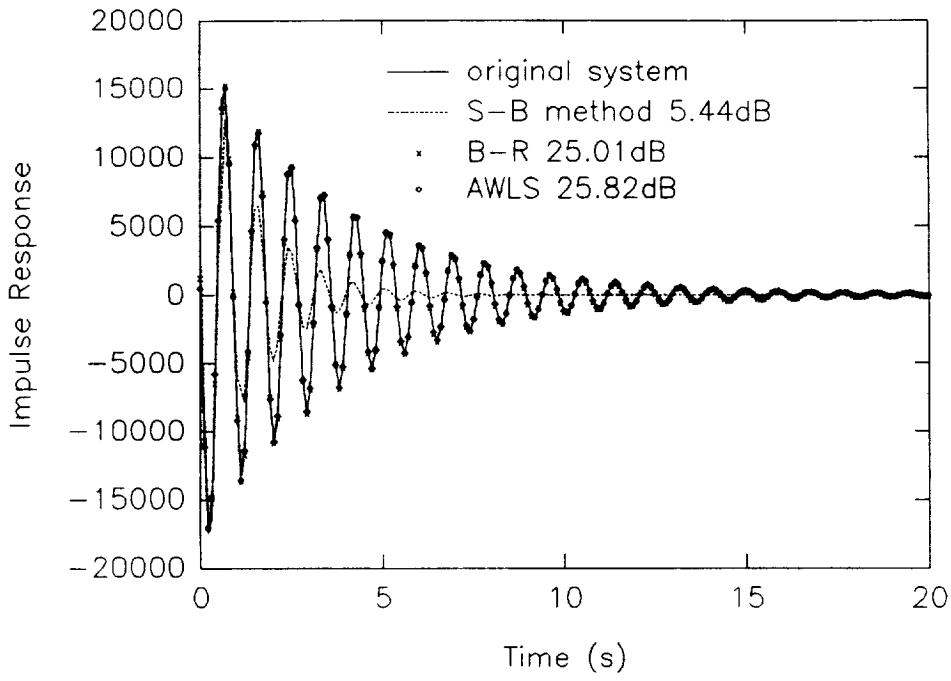

Figure 3.20: Impulse responses of the models.

Balanced-Realization also score significantly higher than the S-B scheme. Between AWLS/MFT and Balanced-Realization, the AWLS/MFT has better SER numbers in both the frequency and time domains, though the graphical difference is almost invisible. Overall, the poor performance of the S-B method is mirrored in front of the AWLS/MFT and Balanced-Realization schemes.

3.4 Concluding Remarks

From the first two examples, AWLS/MFT did show the promising improvement over the MFFM schemes in both accuracy and data source efficiency, especially when high frequency matching is desired. The FF-Padè method has been compared with the

AWLS/MFT and Balanced-Realization algorithms in three examples and has failed to prove that it is a competitive alternative to these two algorithms. Meanwhile, the FF-Padè still faces the possibility of the resulting model being unstable. In the final example, the S-B method was almost a fiasco relative to the results of the AWLS/MFT and Balanced-Realization algorithms. The Balanced-Realization has been utilized in all the examples and it has held quite a good stance in both the time and frequency domains. It is safe to say that the AWLS/MFT algorithm is working at least as well as the Balanced-Realization scheme due to the fact that in the frequency domain the AWLS/MFT has more or less an edge (larger SER numbers) over the Balanced-Realization, while in the time domain they both share a tiny numerical lead one way or another. One interesting point that should be noted regarding the Balanced-Realization scheme is that model reduction is just a very narrow application of the Balanced-Realization technique. Quoting from [24]:

> "the relationship between general model reduction and reduction by subsystem elimination is not well understood".

This somehow is reminiscent of the vagueness of applying AWLS/MFT to model reduction problems. Even though it was developed as a parameter identification algorithm, the AWLS/MFT has shown in the above examples that it can indeed fulfill the mission of model reduction. The Balanced-Realization method takes almost no time to perform the model reduction, while the overall time (including synthesizing)

of AWLS/MFT is ranging from no time to 6 seconds in above examples [4], which does not pose any formidable threat to its practical usage. The limitations of the Balanced-Realization reduction scheme compared with the MFT methods include: (i) although its results fit the low frequency range well, it lacks the flexibility of choosing the frequency range to match models as is possible in the AWLS and MFFM methods; this is demonstrated in the first two examples, and (ii) one premise of applying Balanced-Realization is that the original exact state space or transfer function model must be known beforehand. In this sense, AWLS/MFT algorithm has injected some sort of versatility into the model reduction technique. We have successfully tried to use the Bode diagrams of the above example models instead of analytical high order transfer functions as the start point to approximate these Bode diagrams with specified (low) order models and found that the results are almost identical with those started with high order transfer functions. This underscores the flexibility and versatility of AWLS/MFT in model reduction as one of the major features that could make it stand out among its peers.

[4]This is directly related to the resolving frequency used in reduction. Usually the high frequency matching takes much less time than the low frequency matching reduction.

Chapter 4

Parameter Estimation of MIMO Systems Without Pole Constraints

4.1 Introduction

Consider a multi-input and multi-output (MIMO) continuous time system given by the following transfer function form:

$$\boldsymbol{y}(s) = \boldsymbol{H}(s)\boldsymbol{u}(s) \tag{4.1}$$

where $\boldsymbol{u}(s)$ is a $p_1 \times 1$ transformed input vector, $\boldsymbol{y}(s)$ is a $p_2 \times 1$ transformed output vector, $\boldsymbol{H}(s)$ is a $p_1 \times p_2$ transfer function matrix with $(k,q)^{th}$ element

$$h_{kq}(s) = \frac{B_{kq}(s)}{A_{kq}(s)} \tag{4.2}$$

and $(B_{kq}(s), A_{kq}(s))$ are coprime polynomials in s. Similar to the notation of the SISO cases in Chapter 2, denote θ_{kq} as the parameter vector formed by the coefficients of

the $A_{kq}(s)$ and $B_{kq}(s)$ polynomials, e.g., for an n^{th} order polynomial:

$$A_{kq}(s) = s^n + a_{kq,1}s^{n-1} + \cdots + a_{kq,n}$$

and a $(n-1)^{th}$ degree polynomial:

$$B_{kq}(s) = b_{kq,1}s^{n-1} + \cdots + b_{kq,n}$$

then

$$\theta_{kq} = \begin{pmatrix} -a_{kq,1} \\ \vdots \\ -a_{kq,n} \\ b_{kq,1} \\ \vdots \\ b_{kq,n} \end{pmatrix}. \tag{4.3}$$

Our goal in this Chapter is a systematic procedure of using or extending the modulating function technique (MFT) to estimate the totality of parameter vectors θ_{kq} based on the input-output data pair $\{\boldsymbol{u}(t), \boldsymbol{y}(t)\}$, $t \in [0, T]$. As discussed previously, the order n is presumed chosen beforehand. In practice, this order may be determined by re-solving the identification for increasing orders, starting from an initial value, until the residuals and/or SER's values are sufficiently small and/or large.

4.2 Parameter Identification of MIMO Systems Using AWLS/MFT

4.2.1 Regression Form of Modulated MIMO Systems

Considering the k^{th} output $y_k(s)$, we have

$$y_k(s) = \sum_{q=1}^{p_1} \frac{B_{kq}(s)}{A_{kq}(s)} u_q(s). \tag{4.4}$$

Denote $\tilde{A}_k(s)$ as the least common multiple of $\{A_{k1}(s), ..., A_{kp_1}(s)\}$ and

$$\tilde{B}_{kq}(s) = \frac{B_{kq}(s)}{A_{kq}(s)} \cdot \tilde{A}_k(s) \tag{4.5}$$

Then a modulatable higher order differential operator form of equation (4.4) can be written as

$$\tilde{A}_k(p) y_k(t) = \sum_{q=1}^{p_1} \tilde{B}_{kq}(p) u_q(t) \tag{4.6}$$

where argument p represents the differential operator $p = \frac{d}{dt}$. Using a similar notation and adding the error terms to fit in the identification framework, system (4.1) is then represented in the modulatable differential form:

$$\begin{cases} \tilde{A}_1(p) y_1(t) &= \sum_{q=1}^{p_1} \tilde{B}_{1q}(p) u_q(t) + e_1(t) \\ &\vdots \\ \tilde{A}_{p_2}(p) y_{p_2}(t) &= \sum_{q=1}^{p_1} \tilde{B}_{p_2 q}(p) u_q(t) + e_{p_2}(t) \end{cases} \tag{4.7}$$

Letting $\{\tilde{n}_k\,;\, k = 1, 2, \cdots, p_2\}$ be the corresponding orders of the polynomial set $\{\tilde{A}_k(p)\,;\, k = 1, 2, \cdots, p_2\}$ and $\{\tilde{\theta}_k\,;\, k = 1, 2, \cdots, p_2\}$ the parameter vectors comprising the coefficients of the polynomials $\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\, q = 1, 2, \cdots, p_1\}$ as in (4.3), $k = 1, \cdots, p_2$, apply the complex form of the modulating function set (2.6) of orders

$\{\tilde{n}_k\,;\ k=1,2,\cdots,p_2\}$ successively to the above equations. Notice that these models are overparametrized (in general) due to the cross products of the underlying polynomials comprising the $h_{kq}(s)$. Similar to the SISO case in Chapter 2 (see (2.12), (2.13) and (2.14)), (4.7) could be modulated into the following regression set:

$$\begin{cases} \tilde{Y}_1 &= \tilde{\Gamma}_1\tilde{\theta}_1 + \tilde{\epsilon}_1 \\ &\vdots \\ \tilde{Y}_{p_2} &= \tilde{\Gamma}_{p_2}\tilde{\theta}_{p_2} + \tilde{\epsilon}_{p_2} \end{cases} \tag{4.8}$$

If further we denote by $\{\tilde{M}_k\,;\ k=1,2,\cdots,p_2\}$ the highest modulating frequency index, then each error vector $\tilde{\epsilon}_k$ in (4.8) is of dimension $(\tilde{M}_k+1)\times 1$.

Remark: Here we need to indicate that this problem will be solved as p_2 distinct 2-stage problems: (i) obtain the $\{\tilde{\theta}_k\,;\ k=1,2,\cdots,p_2\}$ for the over-parametrized models, (ii) reduce each over-parameterized model to the original θ_{kq} through the model reduction scheme discussed in Chapter 3.

4.2.2 Joint Likelihood Cost Function

For simplicity, we first look at the cases when $\{\tilde{\epsilon}_k, \tilde{\Gamma}_k, \tilde{Y}_k\,;\ k=1,2,\cdots,p_2\}$ are real[1]. If the error sequences $\{\tilde{\epsilon}_k \sim \mathcal{N}(0,\tilde{W}_k)\,;\ k=1,2,\cdots,p_2\}$ are mutually independent of each other, then the joint likelihood density function can be written as

$$\begin{aligned} \prod_{k=1}^{p_2} p(\tilde{\epsilon}_k|\tilde{\theta}_k) &= \prod_{k=1}^{p_2} \frac{1}{(2\pi)^{(\tilde{M}_k+1)/2}|\tilde{W}_k|^{1/2}} exp\{-\frac{1}{2}\tilde{\epsilon}_k^T\tilde{W}_k^{-1}\tilde{\epsilon}_k\} \\ &= \prod_{k=1}^{p_2} \frac{1}{(2\pi)^{(\tilde{M}_k+1)/2}|\tilde{W}_k|^{1/2}} exp\{-\frac{1}{2}(\tilde{Y}_k-\tilde{\Gamma}_k\tilde{\theta}_k)^T\tilde{W}_k^{-1}(\tilde{Y}_k-\tilde{\Gamma}_k\tilde{\theta}_k)\} \end{aligned}$$

[1]This results by using only the real or imaginary part of the modulating function set to modulate (4.7)

$$= \frac{1}{(2\pi)^{(\tilde{M}_1+\cdots+\tilde{M}_{p_2}+p_2)/2}|\tilde{W}_1|^{1/2}\cdots|\tilde{W}_{p_2}|^{1/2}} exp\{-\frac{1}{2}J(\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2})\}. \quad (4.9)$$

The log-likelihood $\mathcal{L}(\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2})$ can be written as

$$\begin{aligned}\mathcal{L}(\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2}) &= \ln\{\prod_{k=1}^{p_2} p(\tilde{\epsilon}_k|\tilde{\theta}_k)\} \\ &= \kappa(\tilde{W}_1,\cdots,\tilde{W}_{p_2}) - \frac{1}{2}\cdot J(\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2}) \qquad (4.10)\end{aligned}$$

where $J(\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2})$ is a quadratic function

$$\begin{aligned}J(\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2}) &= \sum_{k=1}^{p_2}(\tilde{Y}_k - \tilde{\Gamma}_k\tilde{\theta}_k)^T\tilde{W}_k^{-1}(\tilde{Y}_k - \tilde{\Gamma}_k\tilde{\theta}_k) \qquad (4.11)\\ &\geq 0\end{aligned}$$

and

$$\kappa(\tilde{W}_1,\cdots,\tilde{W}_{p_2}) = -\frac{1}{2}\cdot(p_2 + \sum_{k=1}^{p_2}\tilde{M}_k)\cdot\ln(2\pi) - \frac{1}{2}\sum_{k=1}^{p_2}\ln|\tilde{W}_k|. \qquad (4.12)$$

If the $\{\tilde{W}_k\,;\,k=1,2,\cdots,p_2\}$ are **known**, then maximizing the log-likelihood function (4.10) is equivalent to minimizing the quadratic function $J(\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2})$, and in this case the minimizing parameter set:

$$(\hat{\tilde{\theta}}_1,\cdots,\hat{\tilde{\theta}}_{p_2}) = \arg\min_{\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2}} J(\tilde{\theta}_1,\cdots,\tilde{\theta}_{p_2}) \qquad (4.13)$$

will lead to the maximum likelihood estimate of the parameter set $\{\tilde{\theta}_k\,;\,k=1,2,\cdots,p_2\}$. Regarding the two signal models, i.e., the equation error model and the measurement noise signal model, a few comments could be made here:

1. For the measurement noise signal model **without** input additive noise, i.e., $\{v_q(t)=0; q=1,2,\cdots,p_1\}$ in Figure 4.1, the condition that the $\{\tilde{\epsilon}_k \sim \mathcal{N}(0,\tilde{W}_k)$;

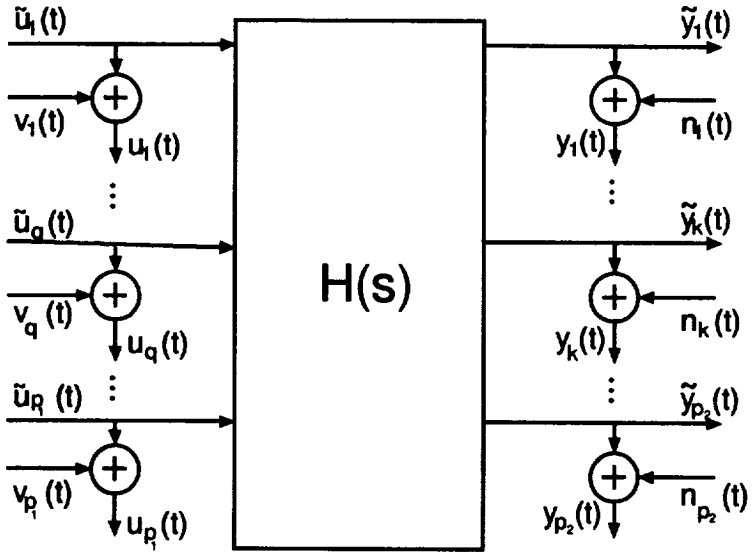

Figure 4.1: MIMO measurement noise signal model.

$k = 1, 2, \cdots, p_2\}$[2] are mutually independent is automatically met; hence, (4.10) is the desired joint cost function. Using Lemma 4 of Section 2.3 which guarantees that the real and imaginary parts of ϵ_k are uncorrelated, the terms in (4.11) can be directly adjusted to the combined forms (see Equations (2.95)~(2.97)) as discussed in the implementation Section 2.3. Neglecting the function $\kappa(\tilde{W}_1, \cdots, \tilde{W}_{p_2})$ for computational expediency, the result from (4.13) is, again, just an approximated maximum likelihood estimate. In reality, the additive noise level in the inputs of many physical systems is significantly lower than that in the outputs. One typical example is the flight data of an F-18 jet aircraft where the inputs include longitudinal pilot stick deflection, lateral pilot stick deflection, horizon-

[2]Similar to Equation (2.45) in the SISO measurement noise signal model, all terms involving the $\{\hat{v}_q(t)\,;\, q = 1, 2, \cdots, p_1\}$ vanish.

tal rudder deflection, horizontal tail deflection and left/right aileron deflections; these are basically free from noise, but the outputs are quite contaminated. Therefore, the focus of our studies will be mainly devoted to this case.

2. When the input additive noises in the measurement noise signal model of Figure 4.1 are non-zero, i.e., $\{v_q \neq 0, q = 1, 2, \cdots, p_1\}$, the error sequences $\{\tilde{\epsilon}_k\,;\, k = 1, 2, \cdots, p_2\}$ are generally correlated and, in this case, it is no longer appropriate to claim that the result is still an approximated maximum likelihood estimate. Nevertheless, the estimate from (4.13) might still be acceptable.

Heretofore, (4.11) will be utilized as a general joint cost function for the MIMO parameter identification problem. As discussed in relation to Lemma 2 and 3 of Chapter 2, the covariance matrices usually can be decomposed as

$$\tilde{W}_k = \sigma_k^2 \underline{\tilde{W}}_k \qquad\qquad k = 1, 2, \cdots, p_2 \tag{4.14}$$

where $\{\sigma_k^2\,;\, k = 1, 2, \cdots, p_2\}$ are the variances of the equation error noise or additive output measurement noise, and $\{\underline{\tilde{W}}_k\,;\, k = 1, 2, \cdots, p_2\}$ are explicitly only related to the binomial coefficients, unknown parameters and modulating frequencies (see Section 2.2.2 and 2.2.3). We assume that the forms or shapes of the probability density functions of the noises are given. In most practical problems, neither the variance of the additive noises nor their probability density functions are known exactly. Usually an assumption can be made about the distribution of noises, but not their variances. In this case, introduce non-negative constants $\{\nu_k\,;\, k = 1, 2, \cdots, p_2\}$ and modify (4.11)

to

$$J(\tilde{\theta}_1, \cdots, \tilde{\theta}_{p_2}) = \sum_{k=1}^{p_2} \nu_k (\tilde{Y}_k - \tilde{\Gamma}_k \tilde{\theta}_k)^T \underline{\tilde{W}}_k^{-1} (\tilde{Y}_k - \tilde{\Gamma}_k \tilde{\theta}_k) \tag{4.15}$$

where the unknown $\{\sigma_k^2\,;\, k=1,2,\cdots,p_2\}$ are absorbed into the $\{\nu_k\,;\, k=1,2,\cdots,p_2\}$. These will be determined along with the parameters in the algorithm to be proposed below.

4.2.3 Decomposability of MIMO Into MISO Models

As indicated in the title of this Chapter, our discussion will be carried out only under the assumption that $\{\tilde{\theta}_k\,;\, k=1,2,\cdots,p_2\}$ are independent one to another and no constraints are attached to them. From the necessary condition of minimizing (4.15):

$$\frac{\partial J(\tilde{\theta}_1, \cdots, \tilde{\theta}_{p_2})}{\partial \tilde{\theta}_k} = -2\nu_k \tilde{\Gamma}_k^T \underline{\tilde{W}}_k^{-1} (\tilde{Y}_k - \tilde{\Gamma}_k \tilde{\theta}_k) = 0 \tag{4.16}$$

we obtain

$$\tilde{\theta}_k = (\tilde{\Gamma}_k^T \underline{\tilde{W}}_k^{-1} \tilde{\Gamma}_k)^{-1} \tilde{\Gamma}_k^T \underline{\tilde{W}}_k^{-1} \tilde{Y}_k. \tag{4.17}$$

This result shows that the kth estimated parameter vector $\tilde{\theta}_k$ is only determined by the output of the kth channel of the MIMO system and by the totality of the inputs $\{u_q(t)\,;\, q=1,2,\cdots,p_1\}$. The important conclusion that can be drawn from the above is that the parameter estimation problem for a p_1-input-and-p_2-output (MIMO) system can be decomposed into a total of p_2 independent MISO sub-problems, if the joint cost function (4.15) is employed.

Under the assumption that the output additive noises $\{n_k(t)\,;\, k=1,2,\cdots,p_2\}$ are mutually independent white Gaussian processes and the input additive noises $\{v_q(t)\,;\, q=$

$1, 2, \cdots, p_1\}$ are of zero variance (or significantly smaller than the variances of the output noises), the AWLS algorithm derived in Chapter 2 can be almost directly applied to obtain an approximate maximum likelihood estimate $\tilde{\theta}_k$ for the kth MISO sub-system $\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\, q = 1, 2, \cdots, p_1\}$. Moreover, only those elements in the parameter vector $\tilde{\theta}_k$ related to the denominator polynomial $\tilde{A}_k(s)$ will be involved in the adaptive iterative procedure.

4.2.4 Reduce from $\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\}$ to $\{A_{kq}(s), B_{kq}(s)\}$; $q = 1, \cdots, p_1$

As mentioned earlier, our ultimate goal is to estimate the coefficients $\{\theta_{kq}\,;\, q = 1, 2, \cdots, p_1\}$ of the original coprime pairs $\{A_{kq}(s), B_{kq}(s)\,;\, q = 1, 2, \cdots, p_1\}$ which are not directly modulatable; instead, we have formulated a joint cost function for estimating the $\tilde{\theta}_k$ for higher order modulatable pairs $\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\, q = 1, 2, \cdots, p_1\}$. Therefore, a model reduction problem will be confronted in the process of going from $\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\, q = 1, 2, \cdots, p_1\}$ to the lower order pairs $\{A_{kq}(s), B_{kq}(s)\,;\, q = 1, 2, \cdots, p_1\}$. Even though there is a little bit of reluctance to admit it, this actually is the very reason that triggered our studies on model reduction problems summarized in Chapter 3. From the six examples in Chapter 3, AWLS/MFT is known to be successful in reducing some weird-looking high order systems into lower order models with accuracy at least equal to the seasoned Balanced-Realization model reduction scheme. As another typical application, the model reduction required from $\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\, q = 1, 2, \cdots, p_1\}$ back to $\{A_{kq}(s), B_{kq}(s)\,;\, q = 1, 2, \cdots, p_1\}$ will be con-

ducted using the AWLS/MFT schemes in our upcoming numerical simulation studies.

4.2.5 Overall Procedure of Estimating MIMO Systems

After these discussions, we propose the following procedure of using the modulating function technique (MFT) to estimate the parameters of a MIMO system, provided the system structure of each $h_{kq}(s)$ be known:

1. In regards to the unmodulatable original MIMO system model (4.4) with its polynomials

   $$\{A_{kq}(s), B_{kq}(s)\,;\; q=1,2,\cdots,p_1\,;\; k=1,2,\cdots,p_2\}$$

   focus on the higher order modulatable MIMO system model (4.6) with its polynomials

   $$\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\; q=1,2,\cdots,p_1\,;\; k=1,2,\cdots,p_2\}$$

   which are obtained by multiplying both sides of (4.4) with the least common multiple polynomial $\{\tilde{A}_k(s)\,;\; k=1,2,\cdots,p_2\}$.

2. Select a set of nonnegative weights $\{\nu_k\,;\; k=1,2,\cdots,p_2\}$, and utilize the joint cost function (4.15) thereby decomposing the MIMO system

   $$\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\; q=1,2,\cdots,p_1\,;\; k=1,2,\cdots,p_2\}$$

   into p_2 sets of MISO sub-systems $\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\; q=1,2,\cdots,p_1\}$.

3. Initialize the index $k = 1$.

4. Apply AWLS/MFT to the kth higher order sub-system $\{\tilde{A}_k(s), \tilde{B}_{kq}(s)\,;\, q = 1,2,\cdots,p_1\}$ and get its approximate maximum likelihood estimate $\hat{\tilde{\theta}}_k$.

5. Using the AWLS/MFT model reduction technique of Chapter 3, convert $\tilde{\theta}_k$ back into $\{\theta_{kq}\,;\, q=1,2,\cdots,p_1\}$ with following steps:

   (a) Inject the q^{th} input $u_q(t)$ into the estimated q^{th} higher order model $\hat{\tilde{h}}_{kq}(s)$ with the MATLAB simulation routine to obtain $\hat{y}_{kq}(t)$, i.e.,

   $$\hat{y}_{kq} = \text{LSIM}(\hat{\tilde{B}}_{kq}, \hat{\tilde{A}}_{kq}, u_q(t), T_{sq}),$$

   Hence, we have acquired an I/O data pair $\{u_q(t), \hat{y}_{kq}(t)\}$ for the SISO system $h_{kq}(s)$.

   (b) Estimate θ_{kq} for the pair $(A_{kq}(s), B_{kq}(s))$ using AWLS/MFT and the I/O data pair $\{u_q(t), \hat{y}_{kq}(t)\}$ from (a). Keep the original algorithm related parameters (ω_0, ω_B) as used in Step 4.

6. $k = k + 1$; if $k > p_2$, stop; otherwise go back to step 4.

4.3 Numerical Simulation Results

Due to the decomposability under the joint cost function $J(\tilde{\theta}_1, \cdots, \tilde{\theta}_{p_2})$, it is only necessary to undertake numerical experiments with MISO systems. Meanwhile for convenience, the index k will be dropped from our previous notations for the MISO subsystems, i.e., modulatable $\{\tilde{A}(s), \tilde{B}_q(s)\,;\, q=1,2,\cdots,p_1\}$ or unmodulatable $\{h_q(s)\,;\, q=$

$1, 2, \cdots, p_1\}$ will be sufficient to represent a MISO system. But first, some potential problems we might face will be stated.

4.3.1 Several Possible Combinations of $\{\tilde{A}(s), \tilde{B}_q(s)\}$

Questions will naturally arise like whether or not large discrepancies in gain and frequency bandwidths among the $\{h_q(s)\,;\, q = 1, 2, \cdots, p_1\}$ will cause some kind of frenzy in terms of accuracy. For the modulating function technique, it could be a major concern that choosing the maximum modulating frequency index M based on the maximum bandwidth among $\{h_q(s)\,;\, q=1, 2, \cdots, p_1\}$ might degrade the estimate for those $h_q(s)$'s with relatively narrow bandwidths. When the only available output is corrupted with additive noise, the impact on the $h_q(s)$'s with smaller gains could be devastating. In order to explore these potential difficult combinations, five 2-input-and-single-output systems $\{h_q(s)\,;\, q=1, 2\}$ will be simulated in our numerical studies. These are configured as follows:

1. Roughly the same gains and frequency bandwidths between $h_1(s)$ and $h_2(s)$;

2. Roughly the same gains but different frequency bandwidths between $h_1(s)$ and $h_2(s)$;

3. Roughly the same frequency bandwidths but different gains between $h_1(s)$ and $h_2(s)$;

4. $h_1(s)$ has a higher frequency bandwidth and a higher gain than $h_2(s)$.

5. $h_1(s)$ has a higher frequency bandwidth but a lower gain than $h_2(s)$.

4.3.2 Setup of Numerical Simulations

Two hundred Monte Carlo runs are going to be carried out at each of several noise levels for each of the five systems mentioned above. The driving inputs $u_1(t)$ and $u_2(t)$ are two Gaussian random sequences generated through the MATLAB routine RANDN() which is basically a white noise generator. Using the linear system simulation routine LSIM(), the outputs of $h_1(s)$ and $h_2(s)$ can be simulated symbolically as

$$y_1(t) = \text{LSIM}(h_1(s), u_1(t), t); \tag{4.18}$$

$$y_2(t) = \text{LSIM}(h_2(s), u_2(t), t); \tag{4.19}$$

Then a single output $y(t)$ available for identification purposes is synthesized through superposing $y_1(t)$ and $y_2(t)$ as

$$\begin{aligned} y(t) &= \underbrace{y_1(t) + y_2(t)}_{\tilde{y}(t)} + n(t) \\ &= \tilde{y}(t) + n(t) \end{aligned} \tag{4.20}$$

where $\tilde{y}(t)$ represents the noise-free (ideal) single output and $n(t)$ is the additive noise formed as

$$n(t) = \mathcal{K} \cdot \text{STD}(y_1(t)) \cdot \text{RANDN}(); \tag{4.21}$$

where the MATLAB routine STD() returns the standard deviation of $y_1(t)$ and the constant $\mathcal{K}$ controls the amplitude of $n(t)$. In order to see the affect of additive noise

on each subsystem, two noise-to-signal ratios (NSR) are defined as

$$\text{NSR}_i = \frac{\|n(t)\|_2}{\|y_i(t)\|_2} \cdot 100\%, \qquad i = 1, 2. \tag{4.22}$$

When $\mathcal{K}$ in (4.21) is zero, which means no additive noise to the single output $y(t)$, the two drivings $u_1(t)$ and $u_2(t)$ will be realized independently for each Monte Carlo run. This kind of simulation will evaluate the performance of the algorithm in rather ideal situations. But another approach will be adopted here if $\mathcal{K} \neq 0$. Thus, $n(t)$ will be realized independently in each Monte Carlo run, while the two inputs $u_1(t)$ and $u_2(t)$, generated independently by RANDN() beforehand, will remain *intact* during the whole ensemble of Monte Carlo simulations. In this way, the noise levels NSR_1 and NSR_2 on $y_1(t)$ and $y_2(t)$ will remain basically unchanged during the 200 Monte Carlo runs. This arrangement not only facilitates our focus on the "reactions" of $h_1(s)$ and $h_2(s)$, but also makes some cross comparisons of the five examples possible. Like before, the time domain performance of the estimated models can be evaluated by the signal-to-error ratio (SER) which is specified as

$$\text{SER} = 20 \cdot \log_{10} \left\{ \frac{\|\tilde{y}(t)\|_2}{\|\tilde{y}(t) - \hat{\tilde{y}}(t)\|_2} \right\} \tag{4.23}$$

where $\hat{\tilde{y}}(t)$ is the simulated output based on the estimated parameters driven by $u_1(t)$ and $u_2(t)$. Note that each SER number will be a random number due to the randomness of $n(t)$ so that its mean and standard deviation will be given as a pair of numbers (m_{SER}, σ_{SER}) in our final numerical tables as determined from the 200 Monte Carlo runs. Let ϱ_k be the kth element of the estimated parameter vector θ and

ϱ_k^o as the kth element of the true parameter vector θ^o. Two additional normalized quantities,

$$\text{normalized bias NB(\%):} \qquad \text{NB} = |\frac{\varrho_k^o - \bar{\varrho}_k}{\varrho_k^o}| \cdot 100\%$$

and

$$\text{normalized standard deviation NSTD(\%):} \qquad \text{NSTD} = |\frac{\sigma_{\varrho_k}}{\varrho_k^o}| \cdot 100\%$$

which characterize the accuracy in parameter space, will be plotted and compared, besides tabulating the mean ($\bar{\varrho}_k$) and variance ($\sigma_{\varrho_k}^2$) for each estimated parameter ϱ_k. For all five systems, the running parameters of MFT are chosen as follows:

- sampling frequency: $f_s = 25.6$Hz
- number of data points: $N = 512$; time interval: $T = 20$s
- resolving frequency: $f_o = 0.05$Hz or $\omega_o = 0.314$(rad/s)
- modulating frequency bandwidth: $f_B = 0.5$Hz or $\omega_B = 3.142$(rad/s)
- maximum modulating frequency index: $M = 10$

4.3.3 Numerical Experiments

Example 1: Consider the two input and single output system

$$\begin{aligned} y(s) &= h_1(s)u_1(s) + h_2(s)u_2(s) \\ &= \frac{5}{s^2+3s+8}u_1(s) + \frac{6}{s^2+4s+10}u_2(s). \end{aligned} \tag{4.24}$$

The Bode diagrams of $h_1(s)$ and $h_2(s)$ are plotted in Figure 4.2 from which we can see that $h_1(s)$ and $h_2(s)$ have rather similar bandwidths and gains. Two hundred Monte Carlo run results are listed in Table 4.1 from which we can see that if there is no additive noise, the algorithm can give an almost perfect estimate, and in this example NSR_1 and NSR_2 are roughly in the same range in each of the different noise levels. The normalized bias (NB) and normalized standard deviation (NSTD)

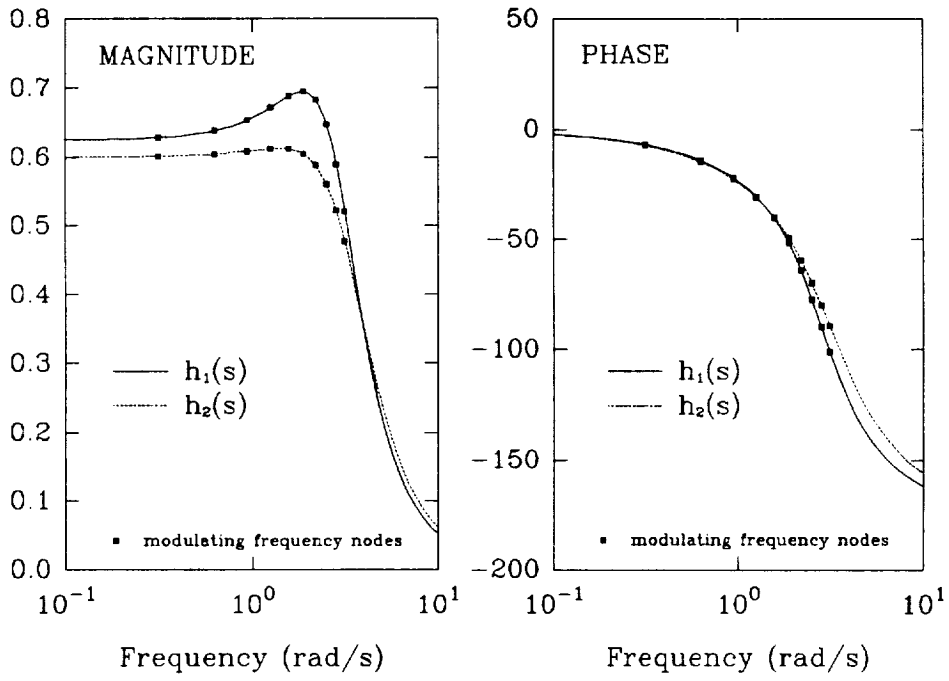

Figure 4.2: Bode diagram of $h_1(s)$ and $h_2(s)$ in Example 1

computed from the table are plotted in Figures 4.3 and 4.4 respectively from which it seems that the influence of the additive noises on $h_1(s)$ and $h_2(s)$ are relatively the same, though $h_1(s)$ has a slightly smaller bias. This case will be used as a reference in the following comparisons.

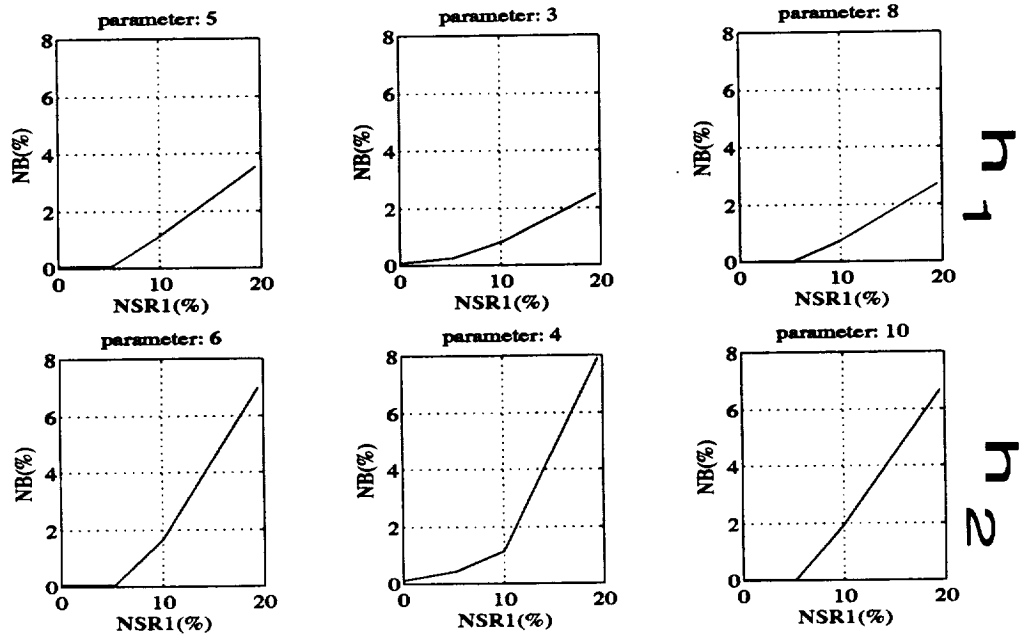

Figure 4.3: Normalized bias vs. additive noise level NSR_1

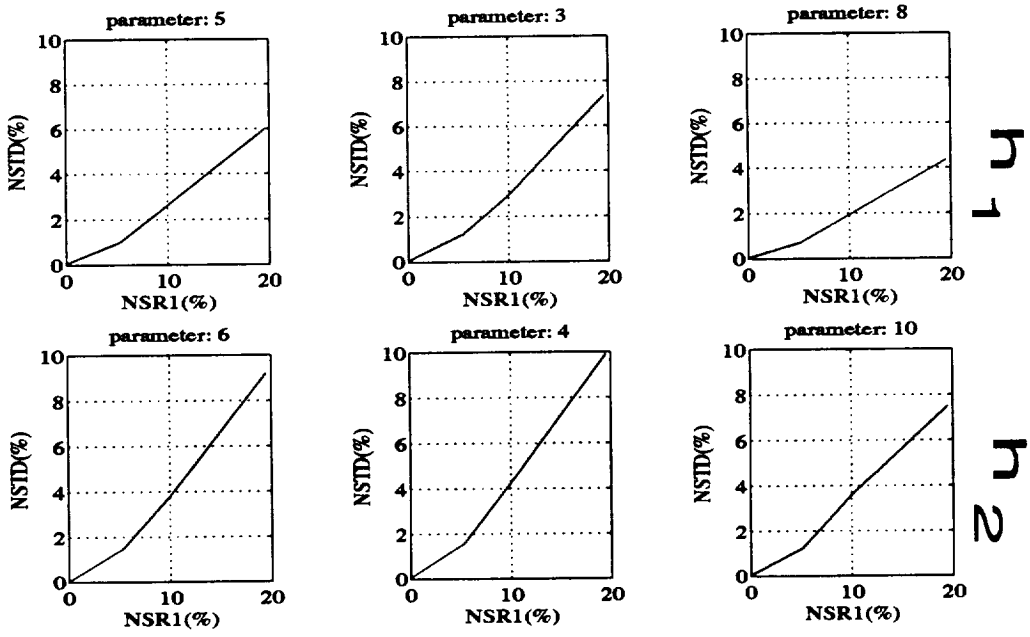

Figure 4.4: Normalized standard deviation vs. additive noise level NSR_1

	$h_1(s)$			$h_2(s)$			NSR_1	m_{SER}
true θ	5	3	8	6	4	10	NSR_2	σ_{SER}
m	4.9957	3.0022	7.9975	5.9965	4.0047	9.9986	0%	57.92dB
σ^2	0.00001	0.00000	0.00001	0.00002	0.00001	0.00002	0%	1.567dB
m	5.0033	3.0078	7.9983	6.0045	4.0179	9.9966	5.22%	42.57dB
σ^2	0.00238	0.00122	0.00341	0.00744	0.00385	0.01615	4.37%	2.912dB
m	5.0581	3.0245	8.0606	5.8998	3.9530	9.8033	9.99%	35.17dB
σ^2	0.01723	0.00770	0.02462	0.05194	0.02939	0.13041	8.367%	3.630dB
m	5.1770	3.0747	8.2181	5.5613	3.6837	9.3342	19.48%	28.11dB
σ^2	0.08889	0.04878	0.12089	0.30429	0.15721	0.56263	16.31%	4.345dB

m: mean value, σ^2: variance, NSR: noise to signal ratio, SER: signal to error ratio.

Table 4.1: Numerical results of 200 Monte Carlo runs for Example 1

Example 2: As the second example, consider the two input and single output system

$$\begin{aligned} y(s) &= h_1(s)u_1(s) + h_2(s)u_2(s) \\ &= \frac{5}{s^2+3s+8}u_1(s) + \frac{3}{s^2+4s+10}u_2(s). \end{aligned} \tag{4.25}$$

The Bode diagrams of $h_1(s)$ and $h_2(s)$ are plotted in Figure 4.5 from which we see that the gain of $h_2(s)$ has been reduced from 0.6 to 0.3 while their bandwidths are still roughly the same. The results of 200 Monte Carlo runs are listed in Table 4.2, and the normalized bias (NB) and normalized standard deviation (NSTD) computed from the table are plotted in Figures 4.6 and 4.7 respectively. Bearing in mind the corresponding relationship between NSR_1 and NSR_2 at each additive noise level in this case

relationship between NSR_1 and NSR_2				
noise	level 1	level 2	level 3	level 4
NSR_1	0%	5.03%	9.62%	21.31%
NSR_2	0%	8.04%	16.10%	35.68%

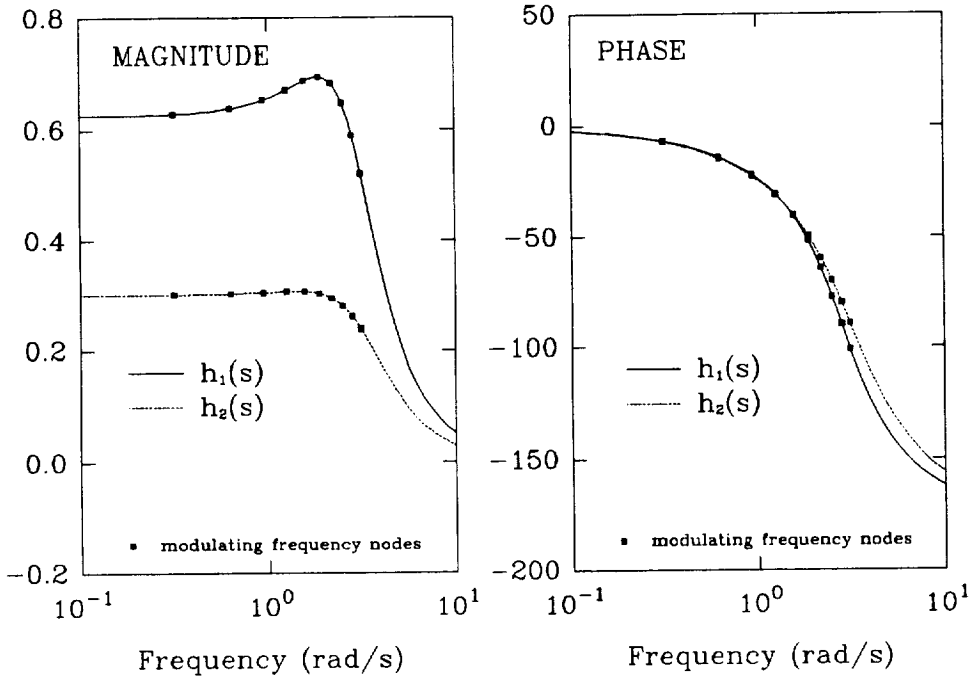

Figure 4.5: Bode diagram of $h_1(s)$ and $h_2(s)$ in Example 2

	$h_1(s)$			$h_2(s)$			NSR_1	m_{SER}
true θ	5	3	8	3	4	10	NSR_2	σ_{SER}
m	4.9959	3.0025	7.9976	2.9980	4.0043	9.9986	0%	57.89dB
σ^2	0.00001	0.00000	0.00001	0.00001	0.00002	0.00005	0%	1.525dB
m	5.0081	3.0122	7.9988	2.9959	4.0144	9.9693	5.03%	42.50dB
σ^2	0.00152	0.00077	0.00255	0.00449	0.00845	0.04110	8.43%	3.089dB
m	5.0574	3.0250	8.0395	2.8461	3.8204	9.4784	9.62%	34.27dB
σ^2	0.00928	0.00506	0.01314	0.03091	0.05747	0.27343	16.10%	4.017dB
m	5.0835	3.0151	8.0728	2.6022	3.4532	8.8351	21.31%	27.51dB
σ^2	0.04719	0.02565	0.07674	0.11289	0.22820	0.79176	35.68%	4.328dB

m: mean value, σ^2: variance, NSR: noise to signal ratio, SER: signal to error ratio.

Table 4.2: Numerical results of 200 Monte Carlo runs for Example 2

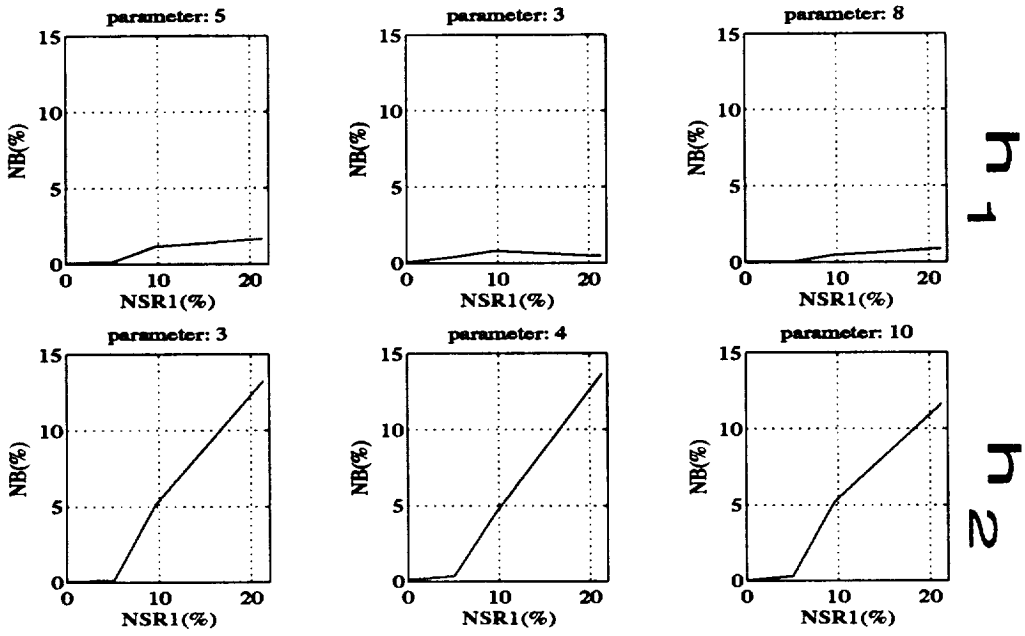

Figure 4.6: Normalized bias vs. additive noise level NSR_1

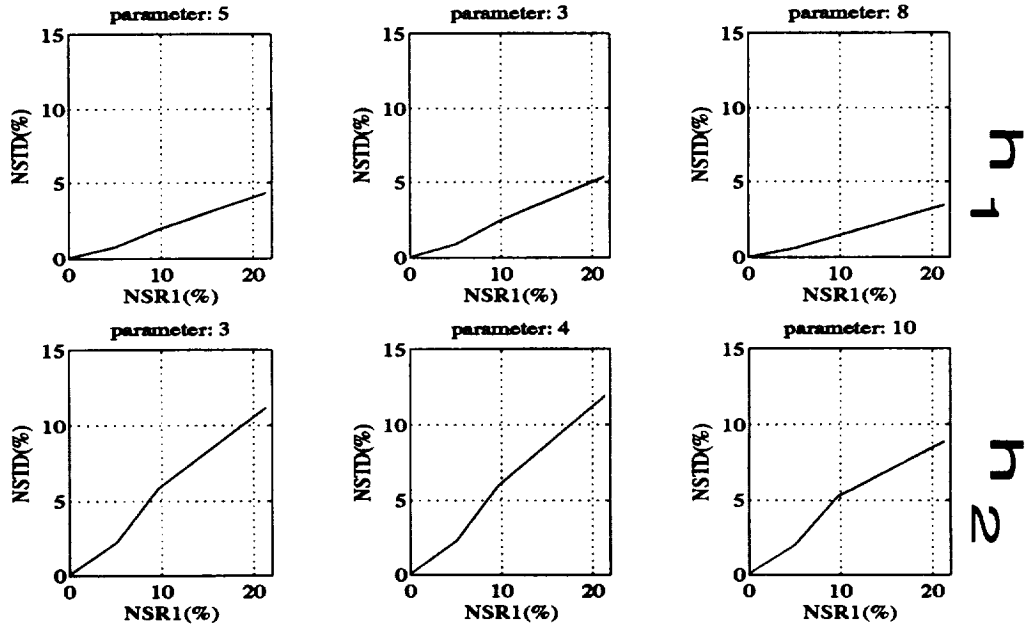

Figure 4.7: Normalized standard deviation vs. additive noise level NSR_1

and comparing with Example 1, we have the following observations: (i) Without additive noise in $y(t)$, both $h_1(s)$ and $h_2(s)$ can be accurately identified; (ii) The NSTD numbers and NB numbers for $h_2(s)$ have increased by $1\% \sim 2\%$ and $3\% \sim 4\%$ respectively at each noise level, which could be ascribed to the relative larger NSR_2; (iii) Note that the corresponding NSTD and NB numbers for $h_1(s)$ have decreased by roughly $1\% \sim 2\%$ at each noise level. These inverse trends for $h_1(s)$ and $h_2(s)$ exaggerate the contrast in the NB and NSTD plots relative to those in Example 1.

Example 3: As the third example, we consider the system

$$\begin{aligned} y(s) &= h_1(s)u_1(s) + h_2(s)u_2(s) \\ &= \frac{5}{s^2+3s+8}u_1(s) + \frac{.6}{s^2+2s+1}u_2(s). \end{aligned} \tag{4.26}$$

The Bode diagrams of $h_1(s)$ and $h_2(s)$ are plotted in Figure 4.8 from which we see

	$h_1(s)$			$h_2(s)$			NSR_1	m_{SER}
true θ	5	3	8	0.6	2	1	NSR_2	σ_{SER}
m	4.9940	3.0014	7.9967	0.5998	1.9997	0.9996	0%	57.74dB
σ^2	0.00001	0.00001	0.00001	0.00000	0.00002	0.00000	0%	2.070dB
m	4.9970	3.0051	7.9911	0.5976	1.9928	0.9967	5.26%	43.34dB
σ^2	0.00120	0.00070	0.00172	0.00017	0.00190	0.00036	9.04%	2.754dB
m	5.0039	3.0070	7.9761	0.5893	1.9654	0.9859	10.02%	37.25dB
σ^2	0.00479	0.00282	0.00703	0.00072	0.00787	0.00144	17.21%	2.823dB
m	4.9973	2.9924	7.9018	0.5575	1.8621	0.9468	19.11%	30.76dB
σ^2	0.01502	0.00918	0.03589	0.00293	0.03144	0.00527	32.83%	2.978dB

m: mean value, σ^2: variance, NSR: noise to signal ratio, SER: signal to error ratio.

Table 4.3: Numerical results of 200 Monte Carlo runs for Example 3

that $h_2(s)$ has a noticeably lower frequency bandwidth while they have roughly the

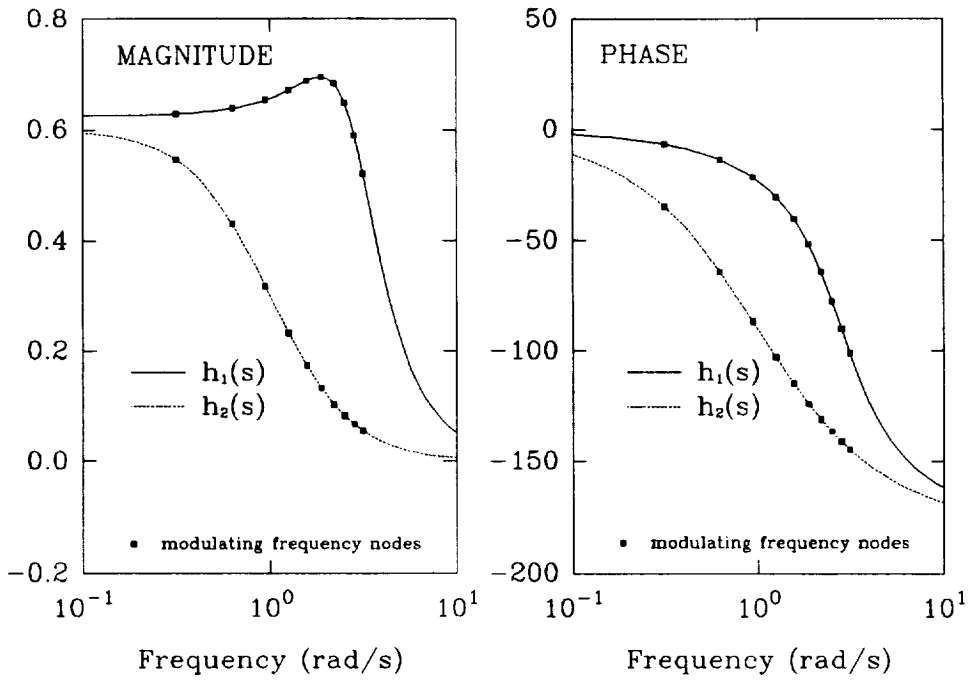

Figure 4.8: Bode diagram of $h_1(s)$ and $h_2(s)$ in Example 3

same steady state gains. The results of 200 Monte Carlo runs are listed in Table 4.3, and the normalized bias (NB) and standard deviations (NSTD) computed from the table are plotted in Figures 4.9 and 4.10 respectively. Again, an almost flawless estimate has been obtained when $\mathrm{NSR}_i = 0$. Because of the narrower bandwidth, the mapping relationship between NSR_1 and NSR_2 is changed to the following:

relationship between NSR_1 and NSR_2				
noise	level 1	level 2	level 3	level 4
NSR_1	0%	5.26%	10.02%	19.11%
NSR_2	0%	9.04%	17.21%	32.83%

Like Example 2, the noise level imposed on $h_2(s)$ is almost doubled. But unlike Example 2, the following two astonishing observations with respect to Example 1 can be made from Figures 4.9 and 4.10: (i) Contrary to intuition, NSTD and NB numbers of $h_2(s)$ are very similar to and even slightly smaller than those in Example 1. This

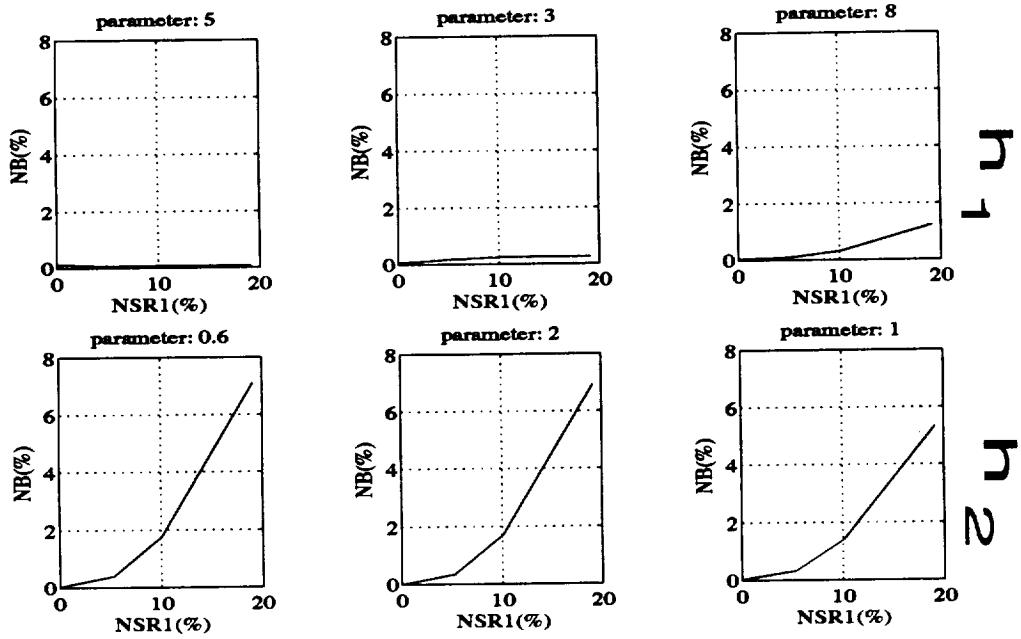

Figure 4.9: Normalized bias vs. additive noise level NSR_1

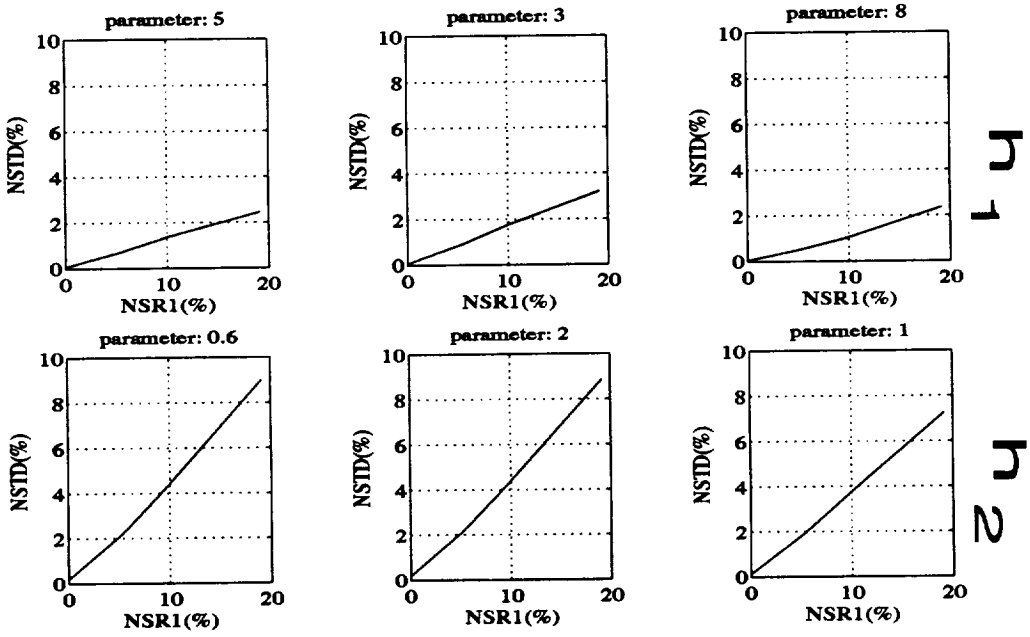

Figure 4.10: Normalized standard deviation vs. additive noise level NSR_1

implies that narrowing the bandwidth of system $h_2(s)$ does not worsen the quality of the estimate for $h_2(s)$ at all, though this narrowing is equivalent to increasing the additive noise to it; (ii) For system $h_1(s)$, its NB values has been *decreased* dramatically and its NSTD numbers are also *down* roughly to *half* of that in Example 1. Another important phenomenon in our Monte Carlo simulation processes is that the iteration steps required for convergence of the AWLS/MFT algorithm in this case are noticeably fewer than those cases in Example 1 and 2 where both bandwidths are close to each other. Apparently the AWLS/MFT is responsible for all of these.

Example 4: As the fourth example, we consider the system

$$\begin{aligned} y(s) &= h_1(s)u_1(s) + h_2(s)u_2(s) \\ &= \frac{5}{s^2+3s+8}u_1(s) + \frac{.3}{s^2+2s+1}u_2(s). \end{aligned} \tag{4.27}$$

The Bode diagrams of $h_1(s)$ and $h_2(s)$ are plotted in Figure 4.11 from which we can

	$h_1(s)$			$h_2(s)$			NSR_1	m_{SER}
true θ	5	3	8	0.3	2	1	NSR_2	σ_{SER}
m	4.9943	3.0018	7.9970	0.2999	1.9998	1.0000	0%	57.11dB
σ^2	0.00001	0.00000	0.00001	0.00000	0.00006	0.00001	0%	1.986dB
m	4.9984	3.0047	7.9895	0.2955	1.9695	0.9889	5.03%	43.52dB
σ^2	0.00078	0.00049	0.00158	0.00014	0.00649	0.00110	17.30%	2.729dB
m	5.0072	3.0065	7.9655	0.2822	1.8799	0.9558	10.10%	36.65dB
σ^2	0.00393	0.00268	0.00693	0.00081	0.03339	0.00586	34.72%	3.322dB
m	4.9888	2.9920	7.8795	0.2504	1.6642	0.8847	20.04%	29.41dB
σ^2	0.02164	0.01136	0.04158	0.00397	0.16048	0.02595	68.85%	3.938dB

m: mean value, σ^2: variance, NSR: noise to signal ratio, SER: signal to error ratio.

Table 4.4: Numerical results of 200 Monte Carlo runs for Example 4

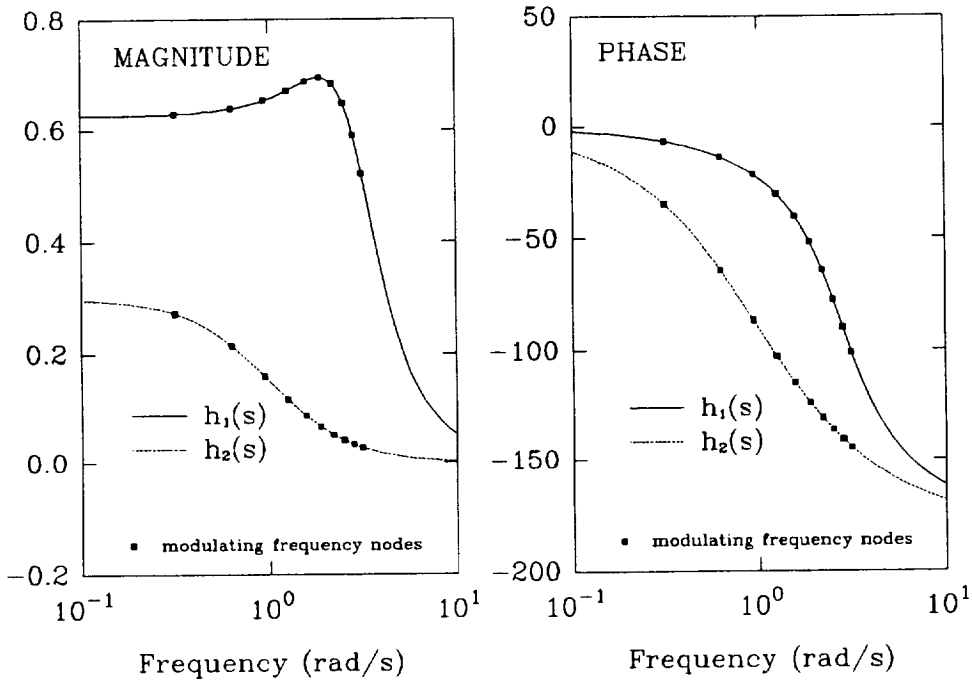

Figure 4.11: Bode diagram of $h_1(s)$ and $h_2(s)$ in Example 4

see that $h_2(s)$ has not only a noticeably lower frequency bandwidth, but also a smaller steady state gain. The results of 200 Monte Carlo runs are listed in Table 4.4, and the normalized bias (NB) and standard deviation (NSTD) computed from the table are plotted in Figures 4.12 and 4.13 respectively. In this case, the correspondence of NSR_1 and NSR_2 is

relationship between NSR_1 and NSR_2				
noise	level 1	level 2	level 3	level 4
NSR_1	0%	5.03%	10.10%	20.04%
NSR_2	0%	17.30%	34.72%	68.85%

Due to the huge amount of equivalent noise imposed on $h_2(s)$, both NB and NSTD values for $h_2(s)$ are approximately doubled comparing to Example 1. Like the results in Example 3 for system $h_1(s)$, the NB numbers are almost diminished to be biasless, and the NSTD values are reduced to only half of those in Example 1. These results

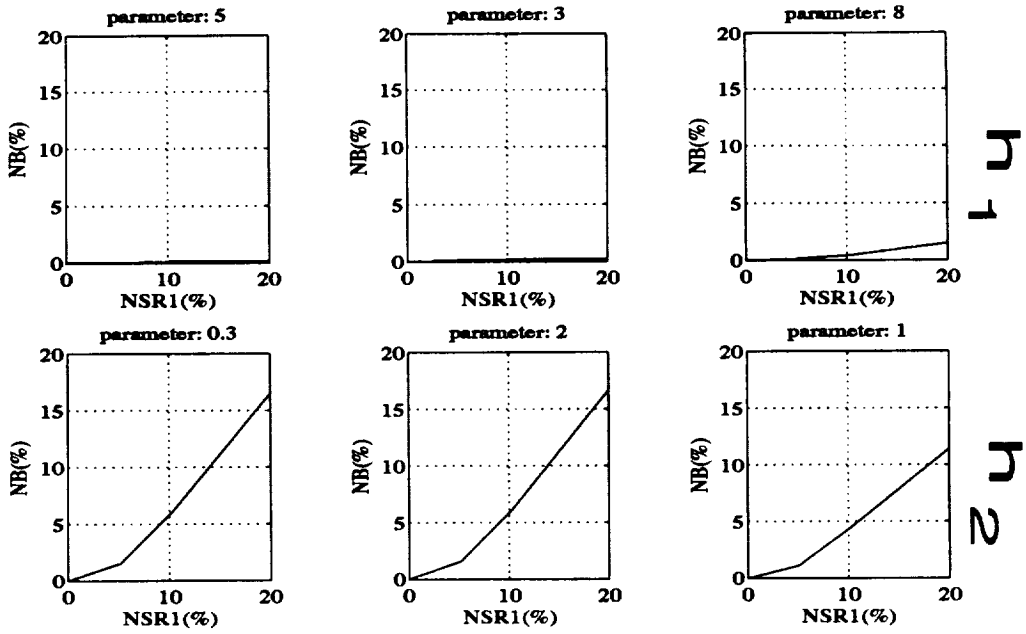

Figure 4.12: Normalized bias vs. additive noise level NSR_1

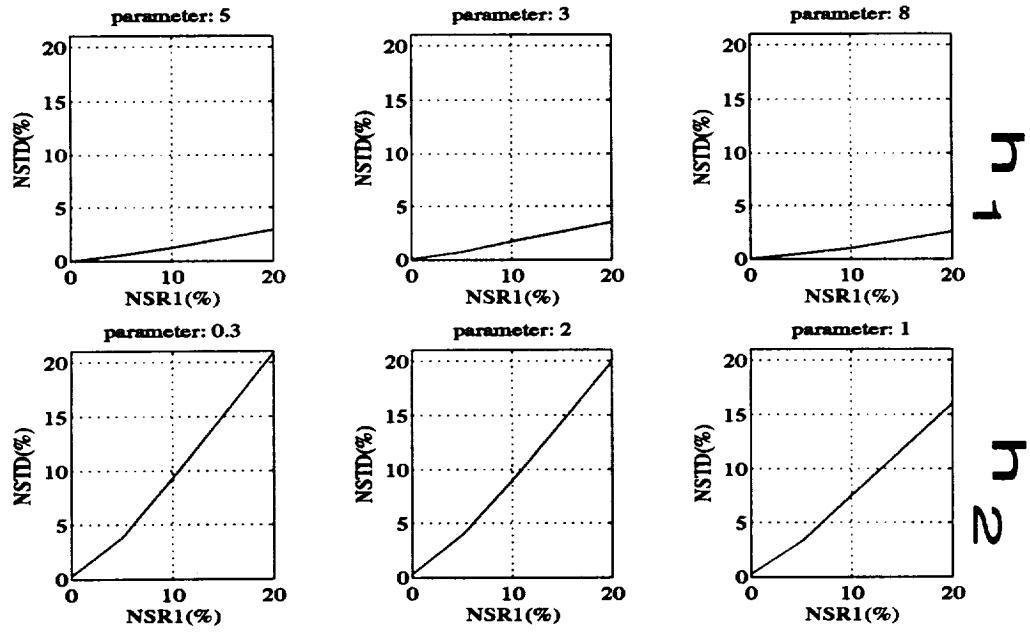

Figure 4.13: Normalized standard deviation vs. additive noise level NSR_1

indicate the sharpest contrast between $h_1(s)$ and $h_2(s)$ among the four examples considered.

Example 5: As the final example, we consider the system

$$\begin{aligned} y(s) &= h_1(s)u_1(s) + h_2(s)u_2(s) \\ &= \frac{5}{s^2+3s+8}u_1(s) + \frac{1.4}{s^2+2s+1}u_2(s). \end{aligned} \tag{4.28}$$

The Bode diagrams of $h_1(s)$ and $h_2(s)$ are plotted in Figure 4.11 from which we

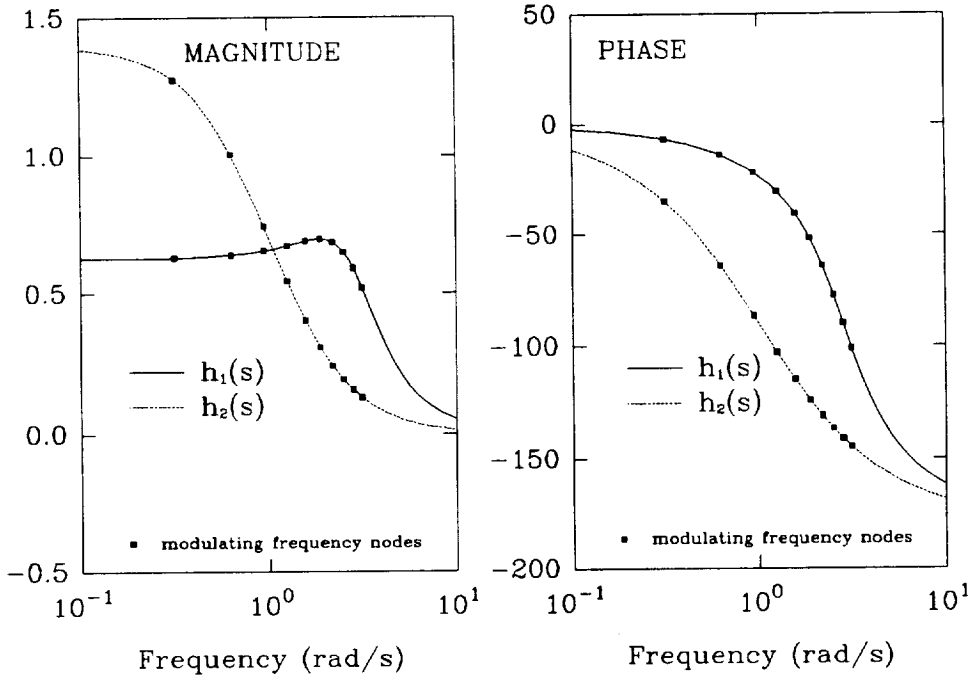

Figure 4.14: Bode diagram of $h_1(s)$ and $h_2(s)$ in Example 5

can see that $h_2(s)$ has not only a noticeably lower frequency bandwidth, but also a higher steady state gain. The results of 200 Monte Carlo runs are listed in Table 4.5, and the normalized bias (NB) and standard deviation (NSTD) computed from the table are plotted in Figures 4.15 and 4.16 respectively. In this case, the corresponding

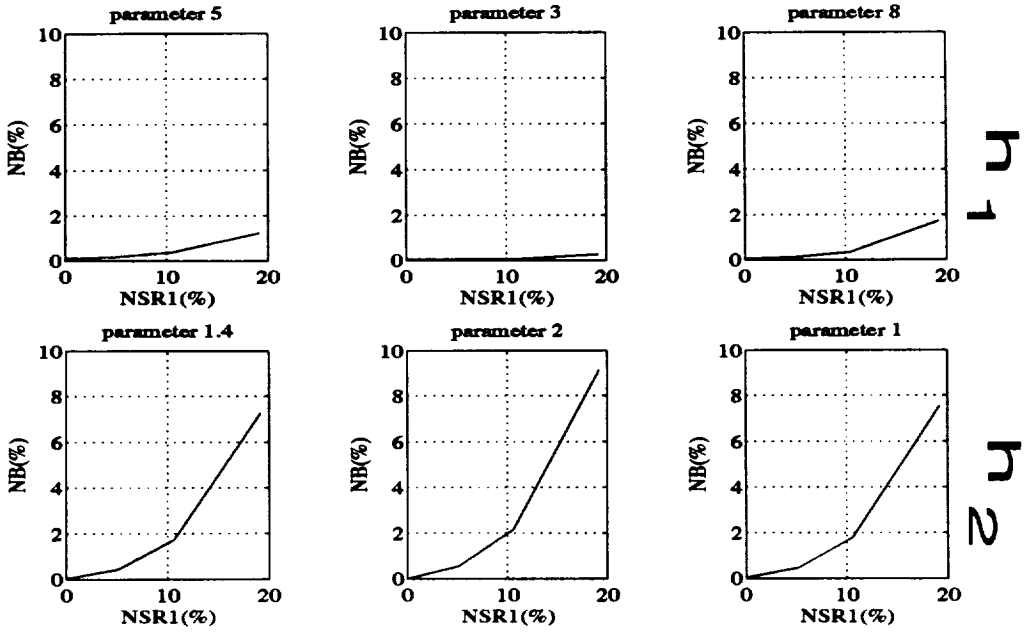

Figure 4.15: Normalized bias vs. additive noise level NSR_1

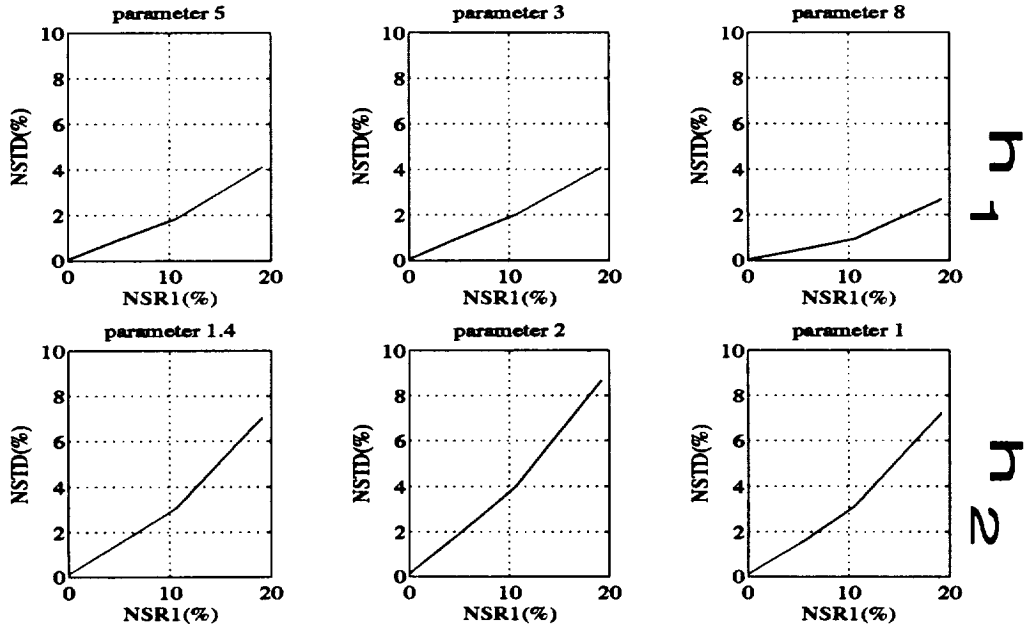

Figure 4.16: Normalized standard deviation vs. additive noise level NSR_1

	$h_1(s)$			$h_2(s)$			NSR_1	m_{SER}
true θ	5	3	8	1.4	2	1	NSR_2	σ_{SER}
m	4.9943	3.0016	7.9971	1.3995	1.9996	0.9996	0%	58.99dB
σ^2	0.00001	0.00001	0.00001	0.00000	0.00001	0.00001	0%	2.297dB
m	4.9908	3.0013	7.9893	1.3937	1.9887	0.9954	5.11%	44.76dB
σ^2	0.00226	0.00097	0.00141	0.00045	0.00156	0.00022	4.63%	3.725dB
m	4.9816	3.0012	7.9708	1.3760	1.9572	0.9821	10.53%	38.24dB
σ^2	0.00824	0.00355	0.00580	0.00183	0.00637	0.00099	9.54%	3.682dB
m	4.9389	2.9919	7.8626	1.2983	1.8174	0.9247	19.12%	30.85dB
σ^2	0.04197	0.01485	0.04599	0.00973	0.03006	0.00521	17.33%	5.112dB

m: mean value, σ^2: variance, NSR: noise to signal ratio, SER: signal to error ratio.

Table 4.5: Numerical results of 200 Monte Carlo runs for Example 5

relationship between NSR_1 and NSR_2 is

relationship between NSR_1 and NSR_2				
noise	level 1	level 2	level 3	level 4
NSR_1	0%	5.11%	10.53%	19.12%
NSR_2	0%	4.63%	9.54%	17.33%

From above form, we can see that additive noise levels in both channels are in the same range, which is analogous to Example 1. However, narrowing the bandwidth of $h_2(s)$ has made the NB and NSTD numbers for $h_1(s)$ much smaller comparing to Example 1. For $h_2(s)$, the NSTD numbers are also decreased and the NB numbers remain basically unchanged with respect to Example 1. Again, the iteration steps required for convergence of the AWLS/MFT algorithm in this case are noticeably fewer than those cases in Example 1 and 2.

4.3.4 Cross Comparisons and Comments

In our arrangement, system $h_1(s)$ and its I/O pairs $(u_1(t), y_1(t))$ have been kept the same in all five examples; therefore, noise generated by equation (4.21) guarantees that the three additive noise levels (about$5\%, 10\%, 20\%$) are physically in the same range in all five examples. After the five examples regarding different SISO subsystem combinations have been presented, some cross comparisons among them will be made here with the hope that further revelation could be made about the influence of different subsystem combinations on estimate bias and deviations.

When there is no additive noise present in the output $y(t)$, the parameters have been perfectly retrieved in all five MISO systems, no matter what combination it has.

A narrower bandwidth as in Examples 3 and 5 has surprisingly improved both the speed and the quality of estimation on both $h_1(s)$ and $h_2(s)$ (especially for system $h_1(s)$). This has not been an anticipated result, because our modulating frequency range has been chosen based on the higher frequency bandwidth, so that the system $h_2(s)$, which has a narrower bandwidth, has been overmodulated with respect to its bandwidth. Pearson and Lee [39] had demonstrated that for the LS/MFT algorithm, this kind of overmodulating could incur estimation error. Therefore, only the contrary was expected at first. These "puzzling" results have been simulated many times and it has been found that the results are very much repeatable. This fact, on the other hand, is helpful in making the following two conclusions: (1) The AWLS/MFT algorithm is not as sensitive to the chosen modulating bandwidth or to

overmodulating as is the LS/MFT scheme[3]; (2) The difference in bandwidth might actually make it easier for the AWLS/MFT algorithm to separate or distinguish the contributions from the two systems $h_1(s)$ and $h_2(s)$ to the single available output, and this might be more or less like a two-category classification problem in a pattern recognition framework in which the greater the distance between two clusters in the feature vector space, the better or more accurate the classification will be.

When a lower gain $h_2(s)$ is used as in Example 2, both the NB and NSTD values for $h_2(s)$ were up slightly with respect to Example 1, but meanwhile the NB and NSTD numbers for $h_1(s)$ came down a little bit, correspondingly. From Example 3 to Example 4, the gain of $h_2(s)$ was reduced from 0.6 to 0.3; here the adverse impact on the higher gain subsystem $h_1(s)$ has not been noticed at all relative to that in Example 3, while the NB and NSTD numbers for $h_2(s)$ have been doubled. Therefore, lowering the gain of a SISO subsystem in the MISO identification problem will lower the accuracy of estimation for that subsystem, mainly because of the increase of its NSR_2 numbers. This is also consistent with Example 5, where increasing the gain of system $h_2(s)$ has improved its NSTD numbers slightly.

Comparing to Example 1, the composite effect of narrowing the bandwidth and lowering the gain of $h_2(s)$ relative to $h_1(s)$ has been demonstrated more revealingly in Examples 4 and 5. In Example 4, the NB and NSTD values for $h_1(s)$ have been decreased to half those in Example 1, and meanwhile these two numbers have been

[3]This property of AWLS/MFT will be further illustrated in the estimation of the Longitudinal dynamics of an F-18 aircraft (see Section 5.4.2).

almost doubled for $h_2(s)$. In Example 5, narrowing the bandwidth and increasing the gain of system $h_2(s)$ not only reduced the NB and NSTD numbers for $h_1(s)$, but also lower the NSTD numbers for system $h_2(s)$.

Another measure of quality of parameter identification is the time domain performance of the estimated systems or models. The SER number mentioned in Section 4.3.2 is one such quantity characterizing the time domain performance of the model. In our Monte Carlo simulation studies, the mean and standard deviation of SER have been recorded and presented in Tables 4.1$\sim$ 4.5 from which the SER numbers in all five examples are fairly close to each other at each noise level NSR_1; consistent with the NB and NSTD measures, Example 3 has about a 2dB edge over the equal bandwidth combinations. In order to provide some visual perception about the meaning of SER number, one typical Monte Carlo run in Example 1 with $\text{NSR}_1 = 20\%$ has been plotted in Figure 4.17.

4.4 Concluding Remarks

Without constraints among the denominator polynomials of a MIMO system, the parameter identification can be decomposed into several independent MISO subproblems due to the joint cost function (4.15) stemming from the joint likelihood function of regression errors. The AWLS/MFT has been the core in forming the overall algorithm or procedure to solving this MISO identification problem. If the original form (4.1) is unmodulatable, it has to be converted into the modulatable

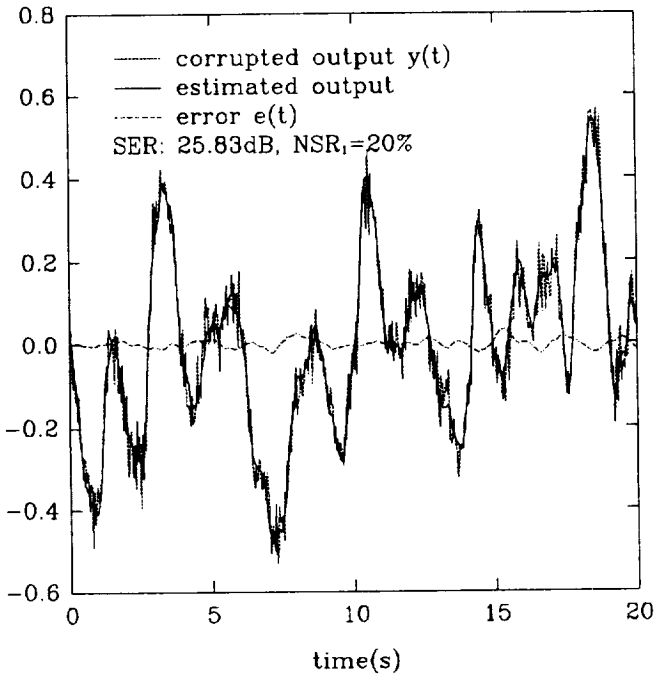

Figure 4.17: A typical realization of $y(t)$, $\hat{\tilde{y}}(t)$, and $e(t) = \hat{\tilde{y}}(t) - \tilde{y}$ from Example 1

form first and then apply the AWLS/MFT algorithm to estimate this higher order modulatable system. After that, as a model reduction tool, the AWLS/MFT scheme with specified lower orders can then be utilized for each subsystem to acquire the parameter identification of the original unmodulatable forms. Due to the fact that model reduction is carried out based on the estimate of the higher order system, it follows that the accuracy of the first AWLS/MFT estimate would be very crucial to the rest of the estimation. With the setup in our simulation studies, the results obtained in this higher order model stage are approximately a maximum likelihood estimate.

The most intriguing phenomenon is the influence of the bandwidths of the two sub-

systems on the final quality of estimation. It is somehow contrary to the conventional wisdom to conclude that this difference of bandwidths actually benefits the estimate for both subsystems, even though the noise level NSR_2 due to the narrowing bandwidth of subsystem $h_2(s)$ has been almost doubled. Some underlying explanation from MFT itself or other approaches like artificial neural networks (ANN) should be targeted in the future research. Another implication of this phenomenon is that the AWLS/MFT algorithm is not sensitive to the modulating frequency bandwidth.

Although the subsystem with lower gain has been seen as a drawback in our examples, the good time domain performance SER values have been persuasive enough to ensure its usability. Overall, the AWLS/MFT algorithm has been successfully applied to MISO parameter identification problems.

Chapter 5

System Identification of the Longitudinal and Lateral Dynamics of an F-18 Aircraft Using the AWLS/MFT Algorithm

5.1 Introduction

5.1.1 Flight Variables Used in This Project

Referring to Figure 5.1, the body axis system of an aircraft is taken with the center at the center of gravity (C.G.) of the airplane, OX forward, OY out the right wing, and OZ downward as seen by the pilot [4]. Most aircraft are symmetric with respect

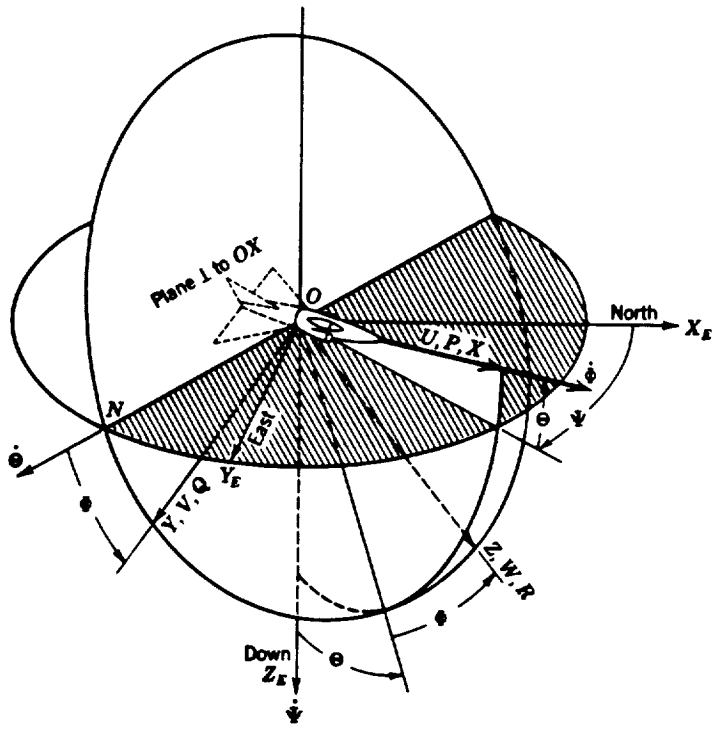

Figure 5.1: Sketch of earth axes and aircraft body axes

to the OXZ plane. In order to describe the motion of the aircraft with respect to the earth or inertial space, a set of Euler angles, specifying the orientation of $OXYZ$ with respect to an earth axis system $OX_EY_EZ_E$ with its origin at the center of gravity of the aircraft and nonrotating with respect to the earth, can be used for this purpose (see Figure 5.1). We denote the three airplane body axis angular velocities as follows:

Q : body axis pitch rate (rad/sec)
P : body axis roll rate (rad/sec)
R : body axis yaw rate (rad/sec).

The transformation between the Euler angles (Ψ, Θ, Φ) and the angular velocities of the aircraft body axis (Q, P, R) can be written as [4]

$$P = \dot{\Phi} - \dot{\Psi}\sin\Theta$$

$$Q = \dot{\Theta}\cos\Phi + \dot{\Psi}\cos\Theta\sin\Phi$$

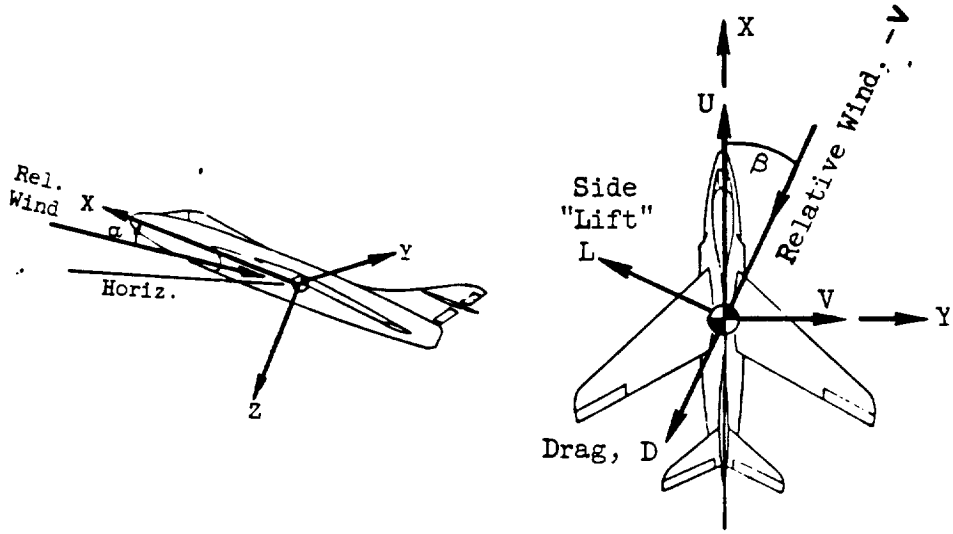

Figure 5.2: Definitions of (1) angle of attack α; (2) sideslip angle β

$$R = -\dot{\Theta}\sin\Phi + \Psi\cos\Theta\cos\Phi$$

Another two output quantities defined in Fig. 5.2 are (1) α : the angle of attack (rad), and (2) β : the sideslip angle (rad) of the aircraft. In addition, three major control variables (driving forces) in the F-18 are

δ_h : symmetric horizontal tail deflection (rad)
δ_a : asymmetric aileron deflection (rad)
δ_r : symmetric rudder deflection (rad)

As mentioned earlier, we just need to be concerned about small perturbations, denoted as $(\alpha(t), \beta(t), q(t), p(t), r(t), \phi(t))$, around the equilibrium (or trim) operating point of flight, $(\alpha_0, \beta_0, Q_0, P_0, R_0, \Phi_0)$[1]. These small disturbances will be bounded with a set of linearized differential equations under a series of assumptions (see page 25 in

[1]As suggested by Dr. V. Klein, all the trim conditions in our studies will be determined by averaging the time domain sequences.

[4]); those coefficients (stability derivatives) of the differential equations will be the targets of this chapter.

5.1.2 Criterion of Model Performance

Far different from the simulation studies where the true parameters can be used luxuriously to evaluate the quality of estimations by forming bias and standard deviations, when the original system is not known beforehand, the only seemingly reasonable alternative left is to compare the time domain performance of estimated models with the physical output. Traditionally, as used in the simulation studies, the signal to error ratio (SER), defined as

$$SER = 20 \cdot \log_{10} \left\{ \frac{\|y(t)\|_2}{\|y(t) - \hat{y}(t)\|_2} \right\} \tag{5.1}$$

could serve the purpose of evaluating the time domain performance, where $y(t)$ is the physical output instead of the "ideal" output (as in the simulation cases) and $\hat{y}(t)$ is the estimated output. Due to the long transient time of the aircraft system, the nonzero initial condition used in the MATLAB simulation routine LSIM() might play an important role in determining $\hat{y}(t)$ after the parameters are estimated, although this unknown initial condition has no impact on the quality of the estimations at all. In order to be as objective as possible to estimate $\hat{y}(t)$, a Luenberger Observer based initial condition (I.C.) estimation scheme was suggested by Pearson (see Appendix C) and it will be used as a standard tool in this chapter.

As requested by NASA, the algorithms developed in this chapter will be tested by

estimating the parameters of theoretical models provided by NASA through Monte Carlo simulations at the suggested additive noise level. In these simulation cases, the performance criteria used in Chapter 4 for the estimated models will be employed without further ado.

5.2 Identification of Longitudinal Dynamics

5.2.1 Longitudinal Dynamical System

As mentioned earlier, the longitudinal dynamics involve the response movement of the aircraft in the symmetric plane OXZ and its control signal. In this case, the input control signal would be the symmetric horizontal tail deflection $\delta_h(t)$ and the responses would be (i) the angle of attack $\alpha(t)$, and (ii) the body axis pitch rate $q(t)$ as shown in the block diagram (see Figure 5.3). Hence this is a single-input-and-two-

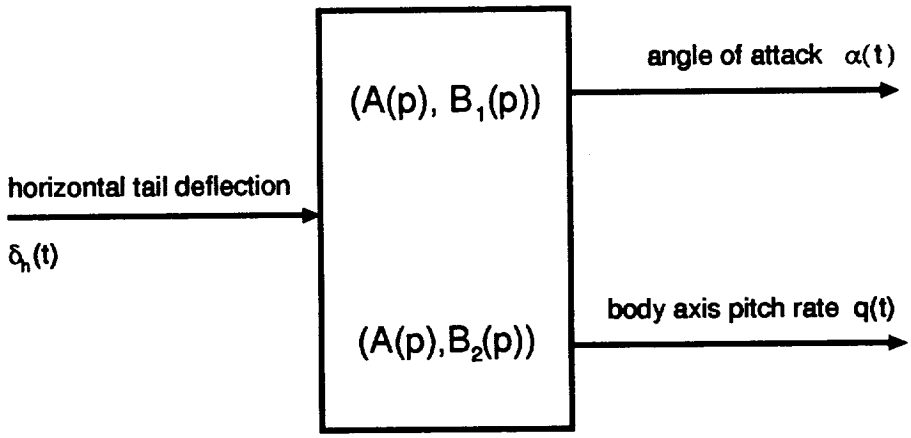

Figure 5.3: Block diagram of the longitudinal dynamics

output (denote as SI2O) system. The general dynamical differential equations are of

the forms[2]:

$$A_1(p)\alpha(t) = B_1(p)\delta_h(t) \tag{5.2}$$

$$A_2(p)q(t) = B_2(p)\delta_h(t) \tag{5.3}$$

$$A_1(p) = A_2(p) = A(p) \tag{5.4}$$

where the orders of $A(p)$, $B_1(p)$ and $B_2(p)$ are expected to be 2, 1 and 1 respectively; more importantly, there exists the physical pole constraint $A_1(p) = A_2(p) = A(p)$[3], which make the algorithms developed in Chapter 4 inapplicable here.

5.2.2 AWLS Algorithm for A Constrained SIMO System

From the flight data received from NASA, the additive noise to $\delta_h(t)$ is basically invisible, while the $\alpha(t)$ and $q(t)$ are contaminated with certain degrees of noise. Applying

[2]Basically this derives from

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad y(t) = Cx(t)$$

where

$$y(t) = \begin{pmatrix} \alpha(t) \\ q(t) \end{pmatrix}, \qquad u(t) = \delta_h(t).$$

According to

$$y(s) = C(sI - A)^{-1}Bu(s),$$

then

$$\det(sI - A)y(s) = C(sI - A)^{-1}\det(sI - A)u(s).$$

[3]Private communication with Dr. V. Klein.

the measurement noise signal model and assuming that the additive noises to $\alpha(t)$ and $q(t)$ are mutually independent white sequences, the joint likelihood cost function, similar to (4.11), can be constructed. Therefore, for the constrained differential dynamical equation set:

$$\begin{cases} A(p)\alpha(t) = B_1(p)\delta_h(t) \\ A(p)q(t) = B_2(p)\delta_h(t) \end{cases} \tag{5.5}$$

consider the joint cost function:

$$J(\theta_1, \theta_2) = (Y_1 - \Gamma_1\theta_1)^T W_1^{-1}(Y_1 - \Gamma_1\theta_1) + \nu(Y_2 - \Gamma_2\theta_2)^T W_2^{-1}(Y_2 - \Gamma_2\theta_2) \tag{5.6}$$

where constant ν is a scaling parameter,

$$\theta_1 = \begin{pmatrix} -\theta_a \\ \theta_{b_1} \end{pmatrix}, \quad \theta_2 = \begin{pmatrix} -\theta_a \\ \theta_{b_2} \end{pmatrix}$$

and θ_a, θ_{b_1} and θ_{b_2} are the parameter vectors comprising the coefficients of $A(p)$, $B_1(p)$ and $B_2(p)$ respectively. Partition the Γ_1 and Γ_2 according to:

$$\Gamma_1 = [\Gamma_{a_1}, \Gamma_{b_1}], \qquad \Gamma_2 = [\Gamma_{a_2}, \Gamma_{b_2}]; \tag{5.7}$$

such that equation (5.6) can be re-written as

$$\begin{aligned} J(\theta_a, \theta_{b_1}, \theta_{b_2}) \;=\; & (Y_1 + \Gamma_{a_1}\theta_a - \Gamma_{b_1}\theta_{b_1})^T W_1^{-1}(Y_1 + \Gamma_{a_1}\theta_a - \Gamma_{b_1}\theta_{b_1}) + \\ & \nu(Y_2 + \Gamma_{a_2}\theta_a - \Gamma_{b_2}\theta_{b_2})^T W_2^{-1}(Y_2 + \Gamma_{a_2}\theta_a - \Gamma_{b_2}\theta_{b_2}) \end{aligned} \tag{5.8}$$

The necessary conditions for minimizing $J(\theta_a, \theta_{b_1}, \theta_{b_2})$:

$$\frac{\partial J}{\partial \theta_a} = 0, \quad \frac{\partial J}{\partial \theta_{b_1}} = 0, \quad \frac{\partial J}{\partial \theta_{b_2}} = 0$$

lead to the following highly coupled equation set for $\{\theta_a, \theta_{b_1}, \theta_{b_2}\}$:

$$(\Gamma_{a_1}^T W_1^{-1}\Gamma_{a_1} + \nu\Gamma_{a_2}^T W_2^{-1}\Gamma_{a_2})\theta_a \;=\; -\Gamma_{a_1}^T W_1^{-1} Y_1 - \nu\Gamma_{a_2}^T W_2^{-1} Y_2 +$$

$$\Gamma_{a_1}^T W_1^{-1} \Gamma_{b_1} \theta_{b_1} + \nu \Gamma_{a_2}^T W_2^{-1} \Gamma_{b_2} \theta_{b_2} \qquad (5.9)$$

$$\Gamma_{b_1}^T W_1^{-1} \Gamma_{b_1} \theta_{b_1} = \Gamma_{b_1}^T W_1^{-1} Y_1 + \Gamma_{b_1}^T W_1^{-1} \Gamma_{a_1} \theta_a \qquad (5.10)$$

$$\Gamma_{b_2}^T W_2^{-1} \Gamma_{b_2} \theta_{b_2} = \Gamma_{b_2}^T W_2^{-1} Y_2 + \Gamma_{b_2}^T W_2^{-1} \Gamma_{a_2} \theta_a \qquad (5.11)$$

Under the assumptions:

- no measurement noise in the input signal $\delta_h(t)$
- the additive noises on $\alpha(t)$ and $q(t)$ are independently Gaussian distributed white processes with the variance ratios embedded in the scaling parameter ν

together with the pole constraint $A_1(p) = A_2(p)$, the weighting matrices W_1 and W_2 are identical, i.e., $W_1 = W_2 = W$, so that updating and inverting W need be computed just once in each iteration. The scaling parameter ν can be used to accommodate the measurement noise difference between the two output channels and to suppress or enhance the importance of the second output in the cost function. The following AWLS/MFT based algorithm has been employed to solve (5.9)~(5.11):

Algorithm 6 (Constrained SIMO AWLS/MFT Algorithm) [4]

1. *Pick a scaling parameter value for ν, $\nu \geq 0$, and estimate the initial value for θ_a through the SISO system model:*

$$A(p)[\alpha(t) + q(t)] = [B_1(p) + B_2(p)]\delta_h(t)$$

using the AWLS/MFT or WLS/MFT algorithm (see Section 2.2.2 and 2.2.3).

[4] Please refer to Section 2.3 for the detailed arrangement of real and imaginary quantities, especially W, W_R and W_I in Equations (2.104) and (2.105), when the AWLS/MFT is implemented.

2. *Compute the weighting matrix W and its inverse (see Section 2.2.2, 2.2.3 and 2.3).*

3. *Substitute the values for the pair $\{\theta_a, W\}$ into* (5.10) *and* (5.11) *and solve for the pair $\{\theta_{b_1}, \theta_{b_2}\}$.*

4. *Estimate a new θ_a from* (5.9) *using the values for $\{\theta_{b_1}, \theta_{b_2}, W\}$ from the previous step.*

5. *Check if the parameter value for the new θ_a has changed or not, based on a percent change in norm. If yes, go back to step 2, otherwise stop.*

6. *Check the system output-signal-to-output error ratios S/E for the two models in* (5.5) *to see if they are in rough agreement with each other. If not, try a new value for ν and repeat steps $1 \sim 6$.*

5.2.3 Results Using Physical Flight Data

Setup of Running AWLS/MFT Scheme

With the sampling rate $F_s = 50$Hz, the longitudinal maneuver flight data totalled 18 seconds (or 900 signal points for each input and output signal). The other running parameters for the SIMO AWLS/MFT Algorithm 6 could be listed as

- resolving frequency: $f_0 = F_s/900 = 0.056$Hz
- modulating bandwidth: $F_b = 0.5$Hz

- highest modulating index: $M = integer(F_b/F_o) = 9$.

As mentioned before, all the DC values of the flight data are subtracted as the trim operating conditions. Therefore, the time averages of all the data blocks (including both longitudinal and lateral channels) displayed in this Chapter are zero.

Identified Longitudinal Dynamical Models

As suggested by a theoretical model of NASA (see appendix A), the order of the model should be $n = 2$. However, two order assumptions, $n = 2$ and $n = 3$, were implemented for comparison purposes and the numerical values of the two estimated models through 20 AWLS/MFT iterations are presented in equations (5.12)~(5.15):

$$n = 2: \qquad \frac{\alpha(s)}{\delta_h(s)} = \frac{-1.4769s - 0.1361}{s^2 + 0.7129s + 0.1804} \tag{5.12}$$

$$\frac{q(s)}{\delta_h(s)} = \frac{0.1265s - 1.2313}{s^2 + 0.7129s + 0.1804} \tag{5.13}$$

$$n = 3: \qquad \frac{\alpha(s)}{\delta_h(s)} = \frac{-1.0381s^2 - 0.5799s + 0.1013}{s^3 + 0.6487s^2 + 0.5507s + 0.0021} \tag{5.14}$$

$$\frac{q(s)}{\delta_h(s)} = \frac{-0.0995s^2 - 1.1270s - 0.5700}{s^3 + 0.6487s^2 + 0.5507s + 0.0021} \tag{5.15}$$

Their time domain performances are plotted in Figures 5.4 and 5.5 respectively. For $n = 2$ and $\nu = 0.1$, the estimated model has $\text{SER}_\alpha = 7.89\text{dB}$, $\text{SER}_q = 9.51\text{dB}$. For $n = 3$ and $\nu = 1$, estimated model has performed even better, especially for channel $\alpha(t)$. In these two cases, 20 iteration steps have been used. The evolution of SER numbers relative to the iteration steps during the process of Algorithm 6 can be

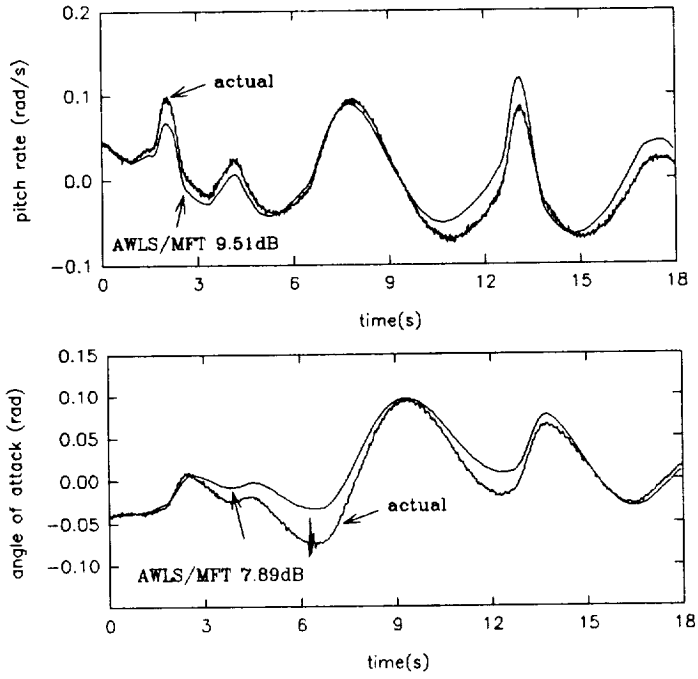

Figure 5.4: Time domain performance of estimated models, $n = 2$

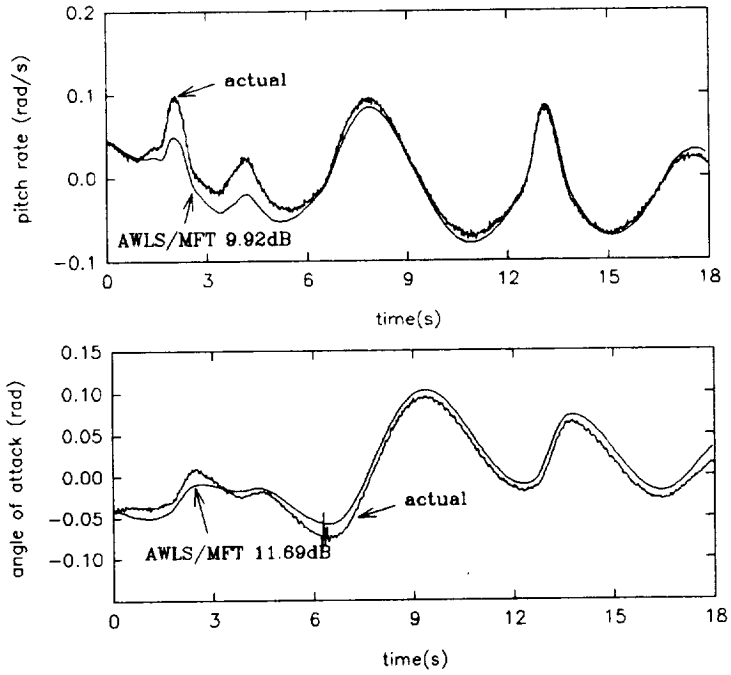

Figure 5.5: Time domain performance of estimated models, $n = 3$

further illustrated through the following discussions and Figure 5.6 where the case of $n = 2$ is plotted.

The Influence of Scaling Factor ν

The overall impact of choosing different ν, originally introduced to accommodate the variance difference of the two additive output noises, can be revealed partly from the SER-vs-iteration curves as typified in Figure 5.6, where a series of ν values and their corresponding curves are displayed. At two extremes, e.g., from $\nu = 10^{-3} \rightarrow 10^{-5}$

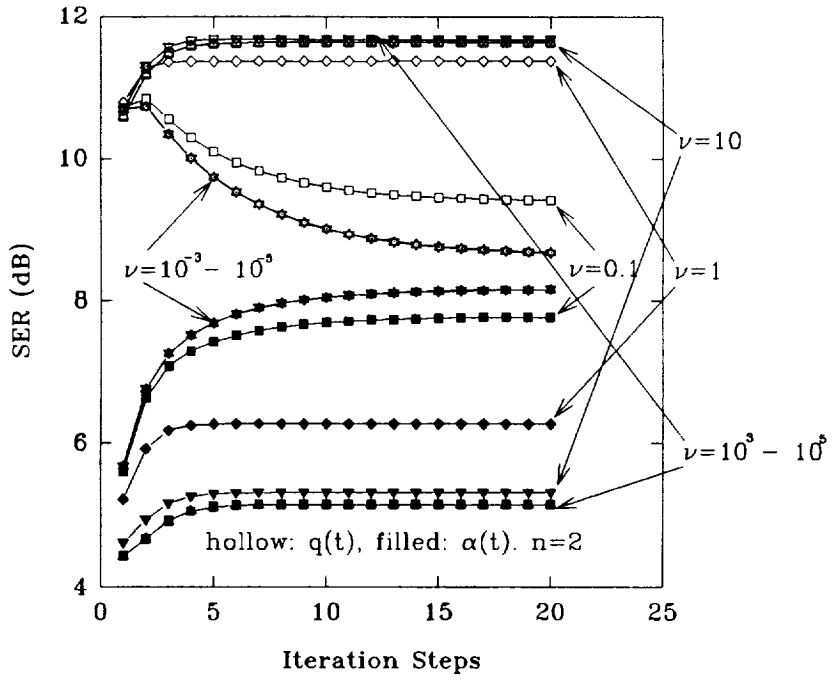

Figure 5.6: The influence of ν on the longitudinal identification

and $\nu = 10^3 \rightarrow 10^5$, the changes of the SER curves are almost invisible as if they have reached two performance bounds. In order to enhance the performance of the $\alpha(t)$ channel, a smaller ν is needed (equivalent to weighting the $q(t)$ channel less in the joint

cost function (5.6)). Not surprisingly, reducing ν will degrade the performance of the $q(t)$ channel. Another factor demonstrating the affect of ν is that in using $\nu = 0.001$, only 5 iteration steps are required to reach $\text{SER}_q = 9.74\text{dB}$ and $\text{SER}_\alpha = 7.69\text{dB}$ (though SER_α and SER_q are still varying rapidly), while 20 iterations are necessary when $\nu = 0.1$ is used. The subtlety of the choice in ν seems far more profound than we have anticipated. The SER trade-off between the $\alpha(t)$ and $q(t)$ channels has been the dictating factor in deciding the ν value in this case. However, this lack of elegancy in determining ν has been far from devastating on account of the general insensitivity to ν, a quick finding for the ν which balances the importance of the two output channels is almost trivial. Exploiting this flexibility will further enable the user to refine the results based on the particular physical setting.

5.2.4 Simulation Results

A numerical simulation study of Algorithm 6 was required by NASA, especially at a set of suggested additive noise levels. Using the theoretical model ($A.2$) of NASA as the true system and driving it with the flight input data (horizontal tail deflection $\delta_h(t)$), 50 Monte Carlo runs are simulated for each of several additive noise levels. Like the previous simulation studies, the white Gaussian noise generator RANDN() is used as the tool to manipulate the noise source. The additive noise levels (standard deviations σ's) suggested by NASA for longitudinal quantities are

- $\sigma_\alpha = 0.1(\text{degree})$

- $\sigma_q = 0.05$(degree/sec)

Although it is a little bit ironic, $\sigma_\alpha = 0.1$(rad) and $\sigma_q = 0.05$(rad/sec), instead of $\sigma_\alpha = 0.1$(degree) and $\sigma_q = 0.05$(degree/sec), were mistaken initially as the standard deviations of the additive noises, which had virtually amplified the original values with a scale 2π. In this case, the additive NSR numbers on $\alpha(t)$ and $q(t)$ had reached an appalling level, 200% and 95% respectively. Fortunately, the performance of Algorithm 6 is still quite promising (see Table 5.1, Figure 5.8 and 5.9). Realizing this embarrassingly erroneous scaling, it was decided that the noise level, $\sigma_\alpha = 0.1$(degree) and $\sigma_q = 0.05$(degree/sec), would be used as a normalizing scale to introduce a "normalized" noise-to-signal ratio (NNSR), so that performance in a broader breadth of additive noise levels (at which the relative additive noise between $\alpha(t)$ and $q(t)$ remains unchanged) could be seen (Figure 5.7). Under this normalizing configuration, the values suggested by NASA would be NNSR = 100% (its regular $\text{NSR}_\alpha = 32\%$ and $\text{NSR}_q = 15\%$). One typical realization of the Monte Carlo runs at NNSR = 100% is presented in Figure 5.8; it is seen that Algorithm 6 can provide nearly perfect results at the NASA suggested noise levels. Even at the devastating extreme NNSR = 628% where $\text{NSR}_\alpha = 200\%$ and $\text{NSR}_q = 95\%$, very impressive SER numbers still can be observed from Figures 5.7 and 5.9.

50 Monte Carlo Runs for Longitudinal Dynamics using constrained AWLS/MFT algorithm. Weight 0.25

	numerator 1		numerator 2		denominator		α	q	NSR_α	
true par	-0.0520	-1.4860	-1.5358	-0.2435	0.5095	0.3579	S/E(dB)	S/E(dB)	NSR_q	NNSR
mean	-0.0568	-1.4842	-1.5333	-0.2407	0.5070	0.3576	39.14	44.92	20%	63%
std	0.01168	0.01279	0.00657	0.01006	0.00716	0.00362	3.091	3.482	9%	
mean*	-0.0458	-1.4862	-1.5355	-0.2478	0.5111	0.3592	34.75	40.96	32%	100%
std*	0.01901	0.01843	0.01127	0.01575	0.01094	0.00629	3.234	2.780	15%	
mean	-0.0531	-1.4898	-1.5356	-0.2497	0.5128	0.3595	31.23	35.96	50%	157%
std	0.02610	0.03085	0.01940	0.03123	0.02062	0.00904	3.948	3.577	24%	
mean	-0.0698	-1.4777	-1.5313	-0.2328	0.4995	0.3556	25.18	30.25	100%	314%
std	0.05507	0.06405	0.03754	0.05004	0.03557	0.01733	3.146	3.191	47%	
mean	-0.0454	-1.5089	-1.5214	-0.2451	0.5123	0.3616	20.00	25.13	200%	628%
std	0.09989	0.13374	0.07999	0.09704	0.06864	0.03403	3.437	3.899	95%	

NNSR is normalized NSR by the case $\sigma_\alpha = 0.1$(deg) {NSR=32%} and $\sigma_q = 0.05$(deg/sec) {NSR=15%}.
*means the conditions suggested by NASA.

Table 5.1: Numerical results of 50 Monte Carlo runs for the longitudinal dynamics

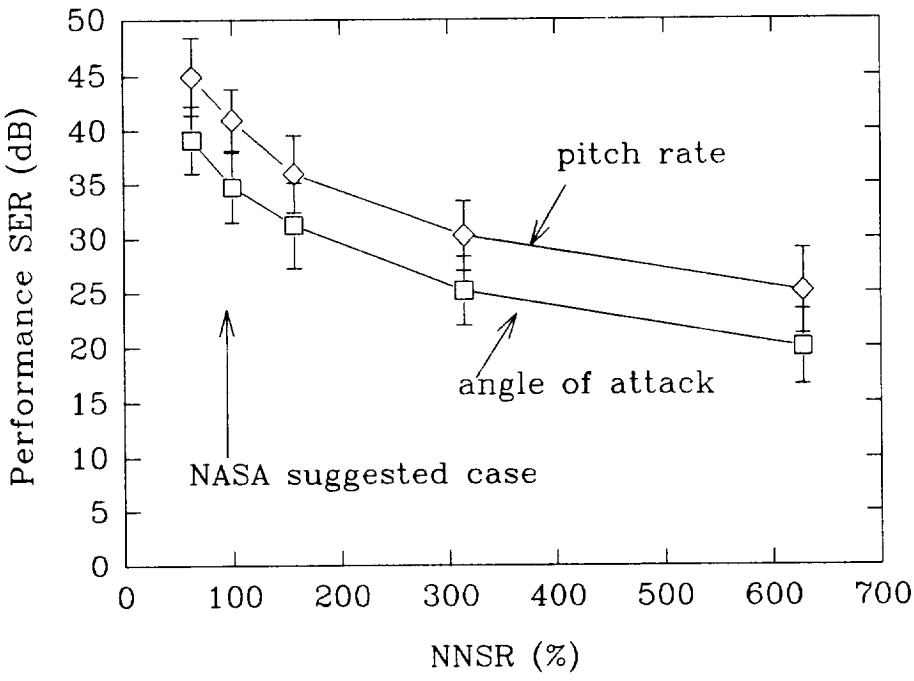

Figure 5.7: SER numbers at different additive noise levels

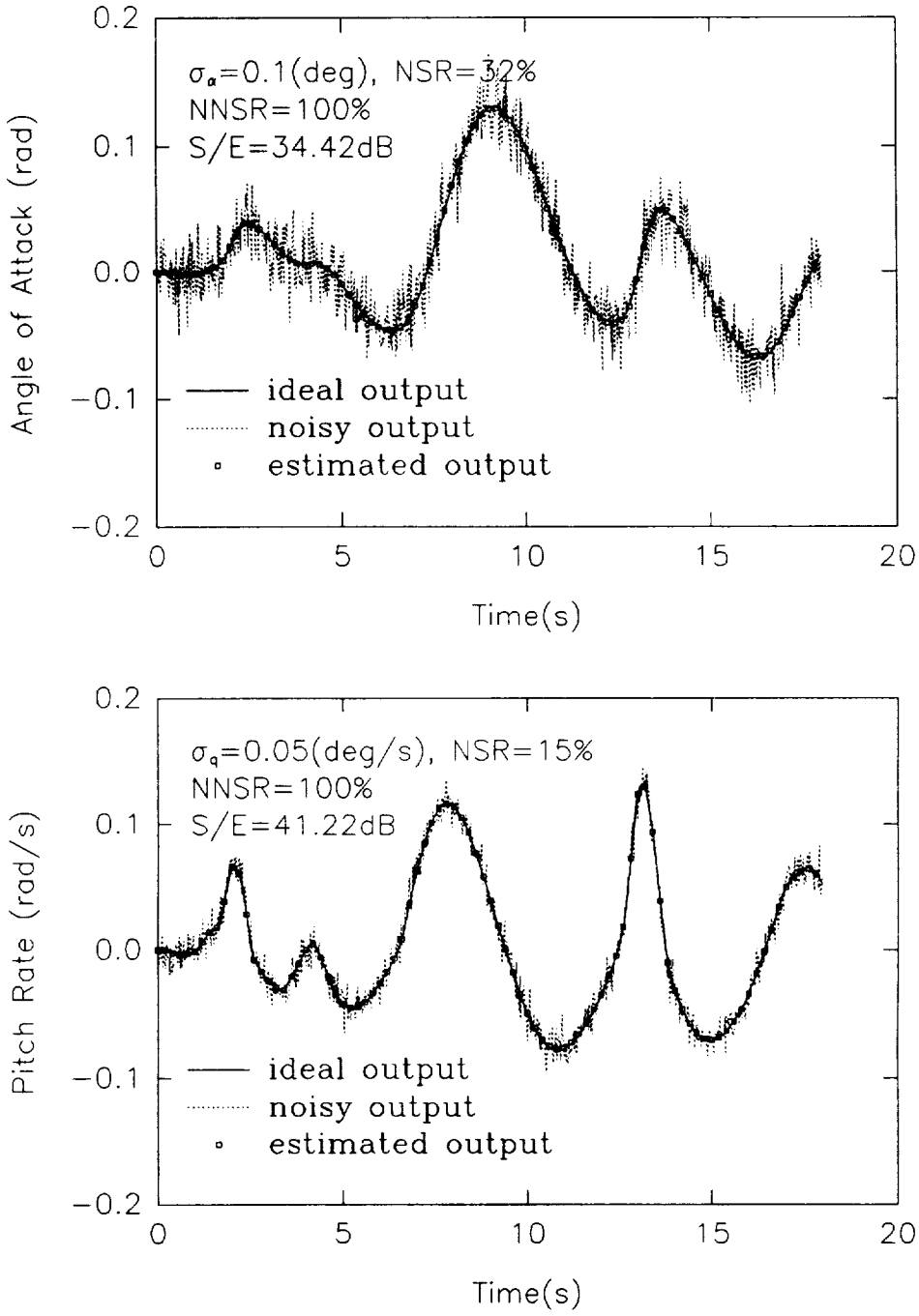

Figure 5.8: Time domain performance of Algorithm 6 at NNSR = 100%

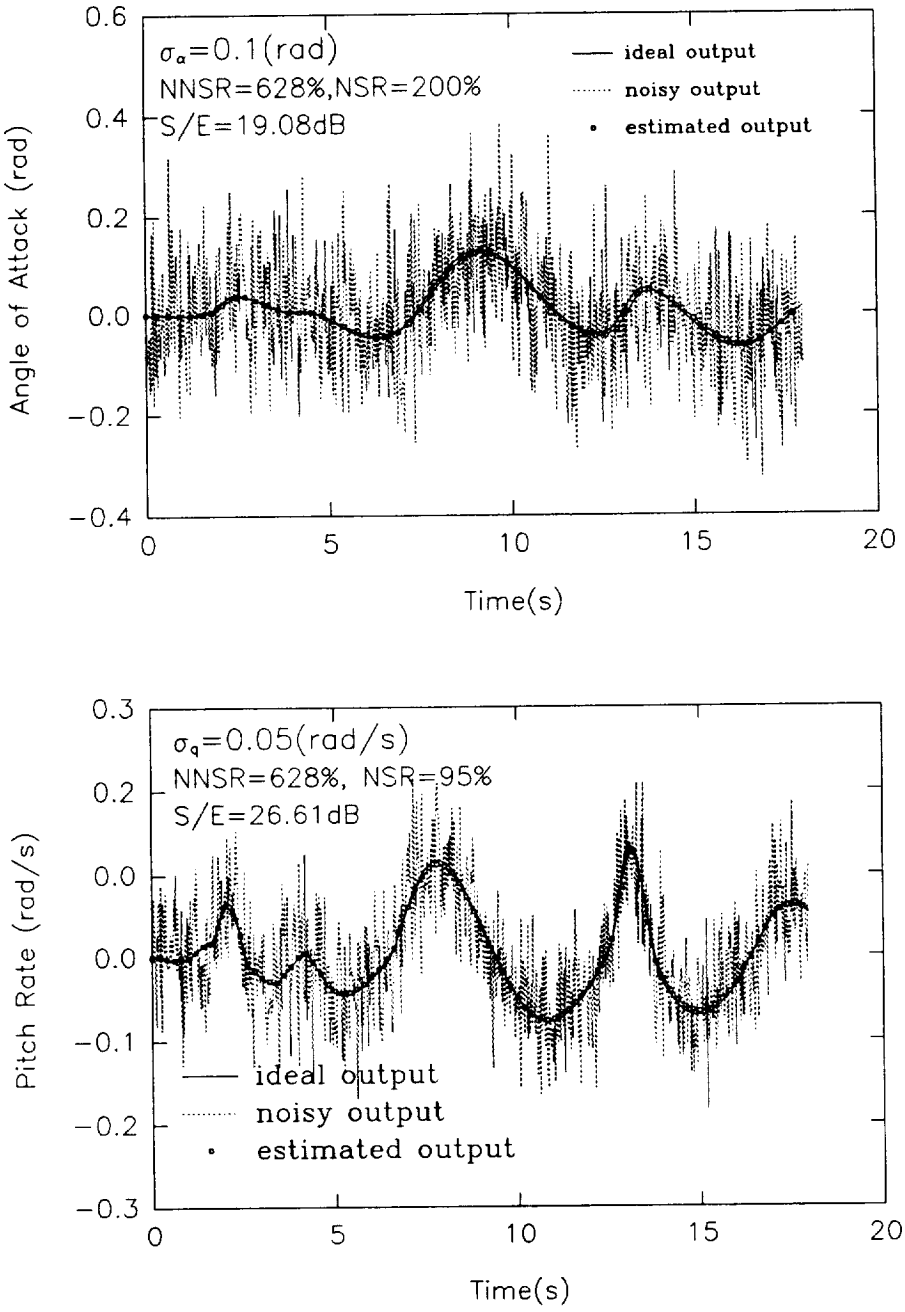

Figure 5.9: Time domain performance of Algorithm 6 at NNSR = 628%

5.3 Identification of Lateral Dynamics

5.3.1 Lateral Dynamical Systems

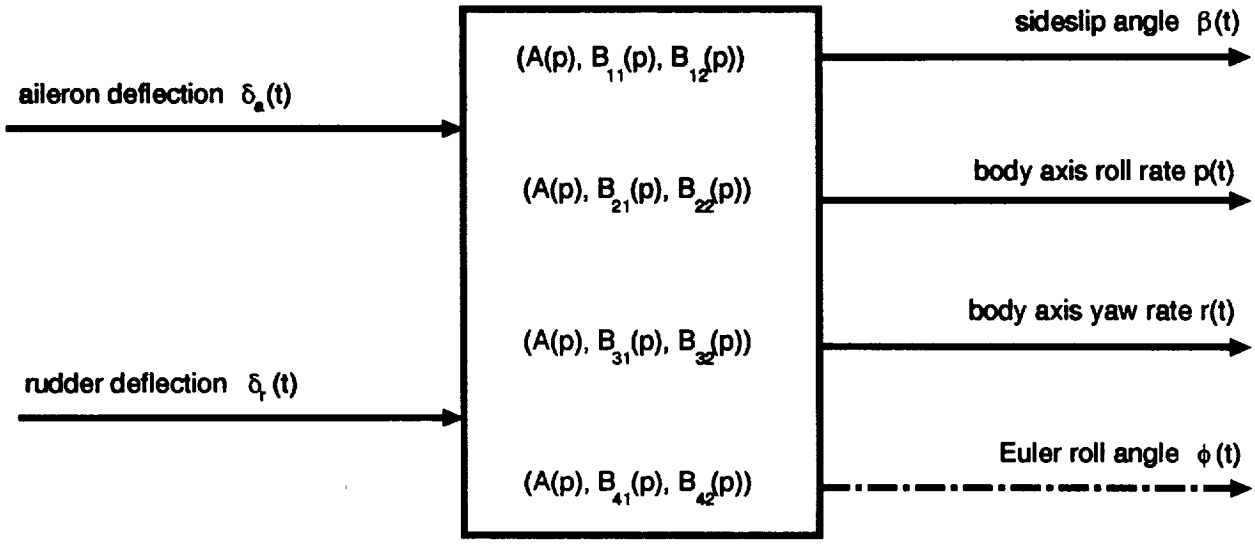

Figure 5.10: Block diagram of the lateral dynamics

The longitudinal dynamics of aircraft seem less complex than the lateral dynamics, which involve more inputs and outputs. As suggested by NASA, three[5] major responses are (1) the sideslip angle $\beta(t)$; (2) the body axis roll rate $p(t)$; and (3) the body axis yaw rate $r(t)$ as plotted in the block diagram (see Figure 5.10). The

[5]In an early communication with NASA, they named these three quantities $(\beta(t), p(t), r(t))$ as the outputs and our research was carried out under this configuration. Later on, from the theoretical model sent by NASA, the Euler roll angle $\phi(t)$ was found to be embedded in the model as well, but it is loosely coupled with the other quantities. In this section, all the analysis will be presented in a two-input-and-four-output (2I4O) framework, from which the 2I3O scheme can be derived trivially.

mathematical differential models describing them can be expressed as

$$\begin{cases} A_1(p)\beta(t) &= B_{11}(p)\delta_a(t) + B_{12}(p)\delta_r(t) \\ A_2(p)p(t) &= B_{21}(p)\delta_a(t) + B_{22}(p)\delta_r(t) \\ A_3(p)r(t) &= B_{31}(p)\delta_a(t) + B_{32}(p)\delta_r(t) \\ A_4(p)\phi(t) &= B_{41}(p)\delta_a(t) + B_{42}(p)\delta_r(t) \end{cases} \tag{5.16}$$

The orders of $\{B_{ij}(p); i = 1,2,3,4; j = 1,2\}$ and $\{A_i(p); i = 1,2,3,4\}$ are expected to be 3 and 4 respectively. The attachment of the physical pole constraint $A_1(p) = A_2(p) = A_3(p) = A_4(p) = A(p)$ to the above differential equation set poses a genuine constrained MIMO system identification problem.

5.3.2 AWLS Algorithm for A Constrained MIMO System

Similar to the longitudinal case, the additive noises to both control signals ($\delta_a(t)$,$\delta_r(t)$) are visually nil. Resorting to the measurement noise signal model and assuming the additive noises to ($\beta(t)$, $p(t)$, $r(t)$, $\phi(t)$) are mutually independent white noises, the joint likelihood cost function like (4.10) of Chapter 4 can be easily formed. Hence, for the constrained dynamical system:

$$\begin{cases} A(p)\beta(t) = B_{11}(p)\delta_a(t) + B_{12}(p)\delta_r(t) \\ A(p)p(t) = B_{21}(p)\delta_a(t) + B_{22}(p)\delta_r(t) \\ A(p)r(t) = B_{31}(p)\delta_a(t) + B_{32}(p)\delta_r(t) \\ A(p)\phi(t) = B_{41}(p)\delta_a(t) + B_{42}(p)\delta_r(t) \end{cases} \tag{5.17}$$

consider the joint cost function:

$$\begin{aligned} J(\theta_1,\theta_2,\theta_3,\theta_4) &= (Y_1 - \Gamma_1\theta_1)^T W_1^{-1}(Y_1 - \Gamma_1\theta_1) \\ &\quad +\nu_1(Y_2 - \Gamma_2\theta_2)^T W_2^{-1}(Y_2 - \Gamma_2\theta_2) \\ &\quad +\nu_2(Y_3 - \Gamma_3\theta_3)^T W_3^{-1}(Y_3 - \Gamma_3\theta_3) \\ &\quad +\nu_3(Y_4 - \Gamma_4\theta_4)^T W_4^{-1}(Y_4 - \Gamma_4\theta_4) \end{aligned} \tag{5.18}$$

where ν_1, ν_2 and ν_3 are the scaling factors. The θ_k vectors can be defined by

$$\theta_1 = \begin{pmatrix} -\theta_a \\ \theta_{b_1} \end{pmatrix}; \quad \theta_2 = \begin{pmatrix} -\theta_a \\ \theta_{b_2} \end{pmatrix}; \quad \theta_3 = \begin{pmatrix} -\theta_a \\ \theta_{b_3} \end{pmatrix}; \quad \theta_4 = \begin{pmatrix} -\theta_a \\ \theta_{b_4} \end{pmatrix},$$

where θ_a, θ_{b_1}, θ_{b_2}, θ_{b_3} and θ_{b_4} are the parameter vectors comprising the coefficients of $A(p)$ and $\{B_{i1}(p), B_{i2}; i = 1,2,3,4\}$ respectively. Partition the Γ_1, Γ_2, Γ_3 and Γ_4 according to

$$\Gamma_1 = [\Gamma_{a_1}, \Gamma_{b_1}]; \quad \Gamma_2 = [\Gamma_{a_2}, \Gamma_{b_2}]; \quad \Gamma_3 = [\Gamma_{a_3}, \Gamma_{b_3}]; \quad \Gamma_4 = [\Gamma_{a_4}, \Gamma_{b_4}],$$

such that equation (5.18) can be re-written as

$$\begin{aligned} J(\theta_a, \theta_{b_1}, \theta_{b_2}, \theta_{b_3}, \theta_{b_4}) \;=\; & (Y_1 + \Gamma_{a_1}\theta_a - \Gamma_{b_1}\theta_{b_1})^T W_1^{-1}(Y_1 + \Gamma_{a_1}\theta_a - \Gamma_{b_1}\theta_{b_1}) \\ & +\nu_1(Y_2 + \Gamma_{a_2}\theta_a - \Gamma_{b_2}\theta_{b_2})^T W_2^{-1}(Y_2 + \Gamma_{a_2}\theta_a - \Gamma_{b_2}\theta_{b_2}) \\ & +\nu_2(Y_3 + \Gamma_{a_3}\theta_a - \Gamma_{b_3}\theta_{b_3})^T W_3^{-1}(Y_3 + \Gamma_{a_3}\theta_a - \Gamma_{b_3}\theta_{b_3}) \\ & +\nu_3(Y_4 + \Gamma_{a_4}\theta_a - \Gamma_{b_4}\theta_{b_4})^T W_4^{-1}(Y_4 + \Gamma_{a_4}\theta_a - \Gamma_{b_4}\theta_{b_4}) \end{aligned} \tag{5.19}$$

The necessary conditions for minimizing $J(\theta_a, \theta_{b_1}, \theta_{b_2}, \theta_{b_3}, \theta_{b_4})$:

$$\frac{\partial J}{\partial \theta_a} = 0, \quad \frac{\partial J}{\partial \theta_{b_1}} = 0, \quad \frac{\partial J}{\partial \theta_{b_2}} = 0, \quad \frac{\partial J}{\partial \theta_{b_3}} = 0, \quad \frac{\partial J}{\partial \theta_{b_4}} = 0$$

lead to the following highly coupled equation set for $\{\theta_a, \theta_{b_1}, \theta_{b_2}, \theta_{b_3}, \theta_{b_4}\}$:

$$\begin{aligned} (\Gamma_{a_1}^T W_1^{-1}\Gamma_{a_1} + \nu_1\Gamma_{a_2}^T W_2^{-1}\Gamma_{a_2} + \nu_2\Gamma_{a_3}^T W_3^{-1}\Gamma_{a_3} + \nu_3\Gamma_{a_4}^T W_4^{-1}\Gamma_{a_4})\theta_a = \\ -\Gamma_{a_1}^T W_1^{-1}Y_1 - \nu_1\Gamma_{a_2}^T W_2^{-1}Y_2 - \nu_2\Gamma_{a_3}^T W_3^{-1}Y_3 - \nu_3\Gamma_{a_4}^T W_4^{-1}Y_4 \\ +\Gamma_{a_1}^T W_1^{-1}\Gamma_{b_1}\theta_{b_1} + \nu_1\Gamma_{a_2}^T W_2^{-1}\Gamma_{b_2}\theta_{b_2} + \nu_2\Gamma_{a_3}^T W_3^{-1}\Gamma_{b_3}\theta_{b_3} + \nu_3\Gamma_{a_4}^T W_4^{-1}\Gamma_{b_4}\theta_{b_4} \end{aligned} \tag{5.20}$$

$$\Gamma_{b_1}^T W_1^{-1}\Gamma_{b_1}\theta_{b_1} = \Gamma_{b_1}^T W_1^{-1}Y_1 + \Gamma_{b_1}^T W_1^{-1}\Gamma_{a_1}\theta_a \tag{5.21}$$

$$\Gamma_{b_2}^T W_2^{-1}\Gamma_{b_2}\theta_{b_2} = \Gamma_{b_2}^T W_2^{-1}Y_2 + \Gamma_{b_2}^T W_2^{-1}\Gamma_{a_2}\theta_a \tag{5.22}$$

$$\Gamma_{b_3}^T W_3^{-1}\Gamma_{b_3}\theta_{b_3} = \Gamma_{b_3}^T W_3^{-1}Y_3 + \Gamma_{b_3}^T W_3^{-1}\Gamma_{a_3}\theta_a \tag{5.23}$$

$$\Gamma_{b_4}^T W_4^{-1}\Gamma_{b_4}\theta_{b_4} = \Gamma_{b_4}^T W_4^{-1}Y_4 + \Gamma_{b_4}^T W_4^{-1}\Gamma_{a_4}\theta_a \tag{5.24}$$

Under the assumptions:

- no measurement noises in the input signals $\delta_a(t)$ and $\delta_r(t)$
- the additive noises on $\beta(t)$, $p(t)$, $q(t)$ and $\phi(t)$ are independent Gaussian distributed white processes with the variance ratios embedded in the scaling parameters ν_1, ν_2 and ν_3

and taking account of the pole constraint $A_1(p) = A_2(p) = A_3(p) = A_4(p)$, the weighting matrices W_1, W_2, W_3 and W_4 are identical, i.e., $W_1 = W_2 = W_3 = W_4 = W$, so that updating and inverting W need be computed just once in each iteration. The scaling parameters ν_1, ν_2 and ν_3 can be used to accommodate the intensity in measurement noise differences among the four output channels and to suppress or enhance the importance of the $p(t)$, $r(t)$ and $\phi(t)$ outputs in the cost function. The following AWLS/MFT based algorithm has been employed to solve (5.20) $\sim$ (5.23):

Algorithm 7 (Constrained MIMO AWLS/MFT Algorithm) [6]

1. *Pick nonnegative scaling parameter values for* (ν_1, ν_2, ν_3), *and estimate the initial* θ_a *through the MISO system model:*

$$A(p)[\beta(t) + p(t) + r(t) + \phi(t)] = [B_{11}(p) + B_{21}(p) + B_{31}(p) + B_{41}(p)]\delta_a(t) + [B_{12}(p) + B_{22}(p) + B_{32}(p) + B_{42}(p)]\delta_r(t)$$

using AWLS/MFT or WLS/MFT algorithm (see Section 2.2.2 and 2.2.3).

2. *Compute the weighting matrix* W *and its inverse (see Section 2.2.2, 2.2.3 and 2.3).*

[6] Please refer to Section 2.3 for the detailed arrangement of real and imaginary quantities, especially W, W_R and W_I in Equations (2.104) and (2.105), when the AWLS/MFT is implemented.

3. *Substitute the values for the pair* $\{\theta_a, W\}$ *into* (5.21) $\sim$ (5.23) *and solve for the parameter set* $\{\theta_{b_1}, \theta_{b_2}, \theta_{b_3}, \theta_{b_4}\}$.

4. *Estimate a new* θ_a *from equation* (5.20) *using the values for* $\{\theta_{b_1}, \theta_{b_2}, \theta_{b_3}, \theta_{b_4}, W\}$ *from the previous step.*

5. *Check if the parameter value for the new* θ_a *has changed or not, based on a percent change in norm. If yes, go back to step 2, otherwise stop.*

6. *Check the system output-signal-to-output-error ratio* S/E *for the three models in* (5.17) *to see if they are in rough agreement with each other. If not, try a new value for the triple* (ν_1, ν_2, ν_3) *and repeat* $1 \sim 6$.

5.3.3 Results Using Physical Flight Data

Setup of Running the AWLS/MFT Scheme

With the sampling rate $F_s = 50$Hz, the lateral flight test data totalled 20.48 seconds long (or 1024 signal points for each I/O channel). The running parameters used for MIMO AWLS/MFT Algorithm 7 are listed below:

- resolving frequency: $f_0 = F_s/1024 = 0.0488$Hz
- modulating bandwidth: $F_b = 0.5$Hz
- highest modulating index: $M = integer(F_b/F_o) = 10$.

Identified Lateral Dynamical Models

In this case, three output channels ($\beta(t)$, $p(t)$, $r(t)$) and two inputs ($\delta_a(t)$, $\delta_r(t)$) are used to form a two-input-and-three-output (2I3O) system with constrained 4th order denominator polynomial. After 22 iterations as shown in Figure 5.11, Algorithm 7 with $\nu_1 = 1$ and $\nu_2 = 1$[7] was truncated and resulted in the following estimated models:

$$\begin{aligned}\beta(s) &= \frac{0.2217s^3 - 1.1464s^2 + 0.5981s - 0.8248}{s^4 + 0.4633s^3 + 2.9945s^2 + 1.0374s + 1.5875} \cdot \delta_a(s) + \\ &\quad \frac{-0.0544s^3 + 0.3066s^2 - 0.1994s + 0.3975}{s^4 + 0.4633s^3 + 2.9945s^2 + 1.0374s + 1.5875} \cdot \delta_r(s) \qquad (5.25)\end{aligned}$$

$$\begin{aligned}p(s) &= \frac{-4.2991s^3 + 0.8664s^2 - 7.8554s + 3.8868}{s^4 + 0.4633s^3 + 2.9945s^2 + 1.0374s + 1.5875} \cdot \delta_a(s) + \\ &\quad \frac{0.4304s^3 - 0.9799s^2 + 0.7119s - 2.5624}{s^4 + 0.4633s^3 + 2.9945s^2 + 1.0374s + 1.5875} \cdot \delta_r(s) \qquad (5.26)\end{aligned}$$

$$\begin{aligned}r(s) &= \frac{-0.3512s^3 + 1.0683s^2 - 1.2603s + 2.5927}{s^4 + 0.4633s^3 + 2.9945s^2 + 1.0374s + 1.5875} \cdot \delta_a(s) + \\ &\quad \frac{-0.1687s^3 - 0.4299s^2 - 0.2223s - 1.1016}{s^4 + 0.4633s^3 + 2.9945s^2 + 1.0374s + 1.5875} \cdot \delta_r(s) \qquad (5.27)\end{aligned}$$

Approximately 3 minutes computation time was needed to calculate these models using an IBM 486/33 machine. The poles of the above models are located at $(-0.1312 \pm 1.3899i)$ and $(-0.2235 \pm 0.8010i)$. The time domain performance of the above models are plotted in Figure 5.12. As pointed out in the NASA model (Appendix B), the fourth state ($\phi(t)$) could be used as a fourth output. The identification for this configuration (equivalent to a 2I4O system like (5.17) was implemented as well by using the Algorithm 7; notice that the resulting SER numbers for the three channels ($\beta(t)$, $p(t)$, $r(t)$) slipped from $(9.41, 11.62, 11.60)$dB to the current

[7]Other ν values have been tested and, like the longitudinal case, the results are not sensitive to the change of ν.

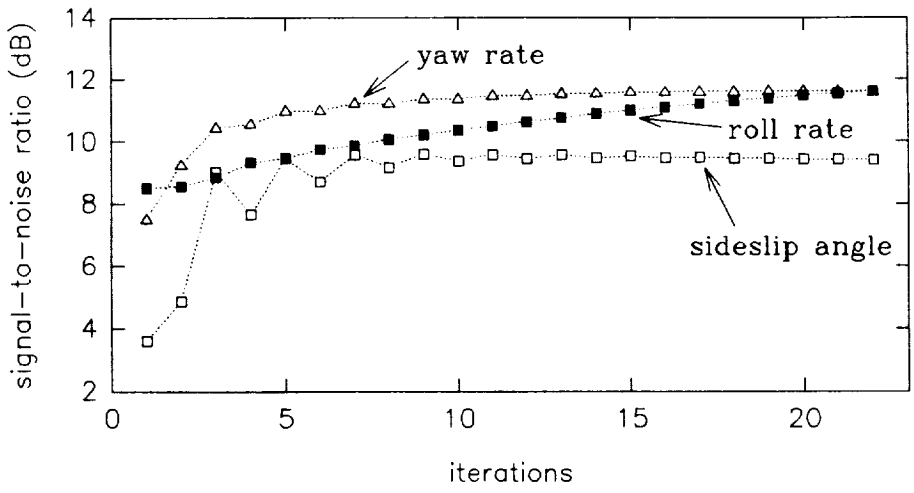

Figure 5.11: SER numbers vs. iteration steps for the lateral dynamics

(7.80, 6.50, 10.44)dB with $\nu_1 = 1$, $\nu_2 = 1$ and $\nu_3 = 1$. However, it should not be too surprising to see that the estimated 2I3O linear models are capable of interpreting the physical data better than those identified with a theoretically linearized 2I4O model structure.

5.3.4 Simulation Results

Similar to the longitudinal case, numerical simulation studies on Algorithm 7 were requested by NASA at a set of suggested additive noise levels. Using the theoretical model ($B.2$) of NASA as the true system and driving it with the recorded flight input data ($\delta_a(t)$ and $\delta_r(t)$), 50 Monte Carlo runs are simulated for each additive noise level on this 2I4O system. Like the previous simulation studies, the white Gaussian noise generator RANDN() is used as the tool to manipulate the noise source. The additive noise levels (standard deviation σ's) suggested by NASA for the lateral dynamics are

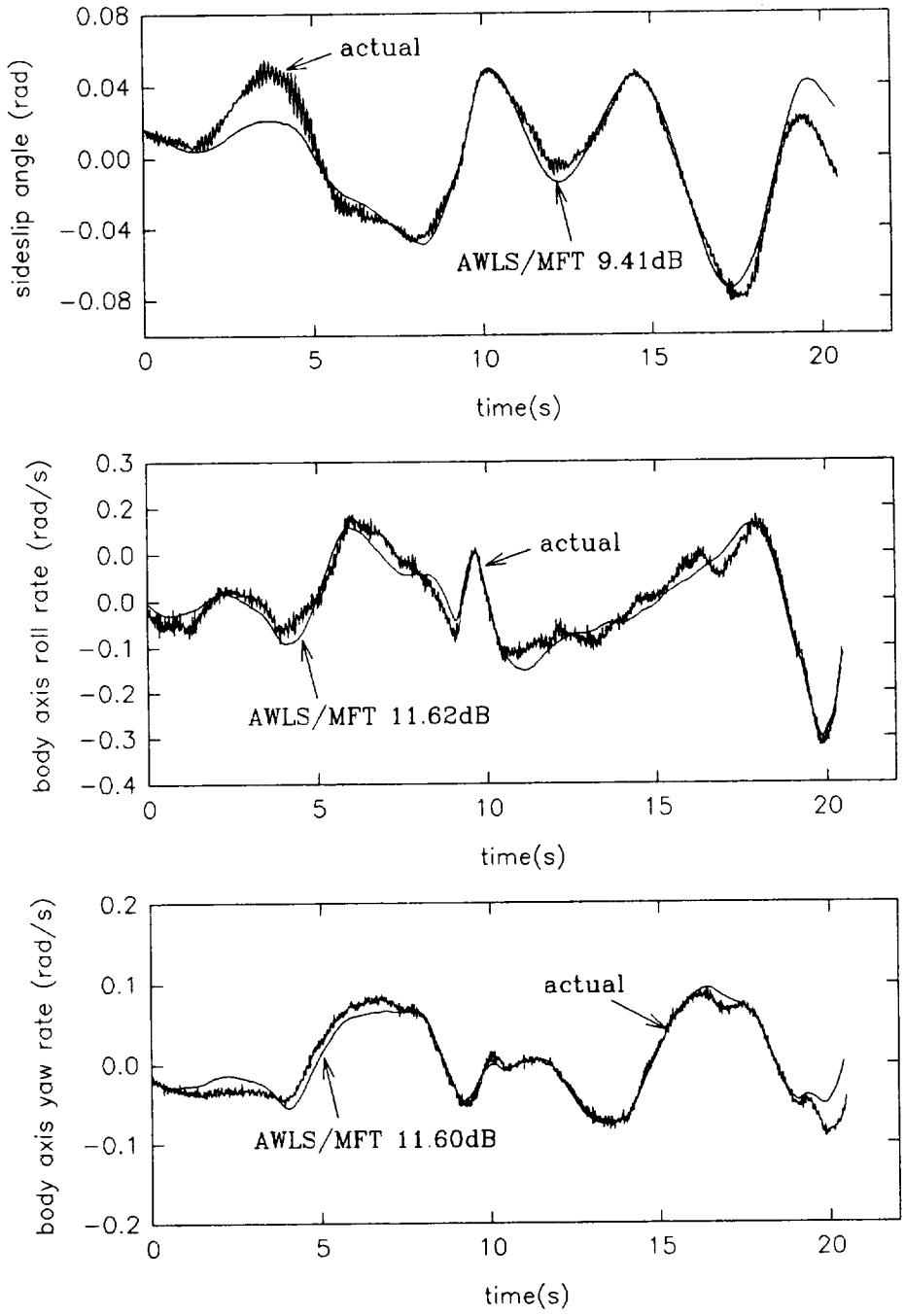

Figure 5.12: Time domain performance of estimated lateral models

- $\sigma_\beta = 0.1$(degree)
- $\sigma_p = 0.05$(degree/sec)
- $\sigma_r = 0.05$(degree/sec)
- $\sigma_\phi = 0.01$(degree).

Due to the initial misusage of unit (rad), we actually have tested Algorithm 7 at other larger additive noise settings. In order to represent those results as well, the NASA suggested noise level, the $\sigma_\beta = 0.1$(degree), $\sigma_p = 0.05$(degree/sec), $\sigma_r = 0.05$(degree/sec) and $\sigma_\phi = 0.01$(degree), will be used as a normalizing scale to form the normalized noise-to-signal ratio (NNSR). Then, the NASA suggested noise level would be equivalent to NNSR = 100% ($\text{NSR}_\beta = 51\%$, $\text{NSR}_p = 8\%$, $\text{NSR}_r = 14\%$, and $\text{NSR}_\phi = 1\%$). As mentioned earlier, the Euler angle $\phi(t)$ was excluded from our initial studies on a constrained 2I3O system. Here in the simulation study, the underlying system is known to be the linear 2I4O system ($B.2$). The AWLS/MFT algorithms for both the 2I3O and 2I4O models have been tested for this ideal 2I4O system; their resulting numerical values are listed in Tables 5.2$\sim$ 5.3. As plotted in Figure 5.13, except for the the yaw rate $r(t)$ at the low noise side, the time domain performances of the two models are fairly close to each other. One typical realization for the 2I4O model at the NASA suggested noise level is plotted in Figure 5.14, which confirms that the identified model from Algorithm 7 does give a very good time domain performance.

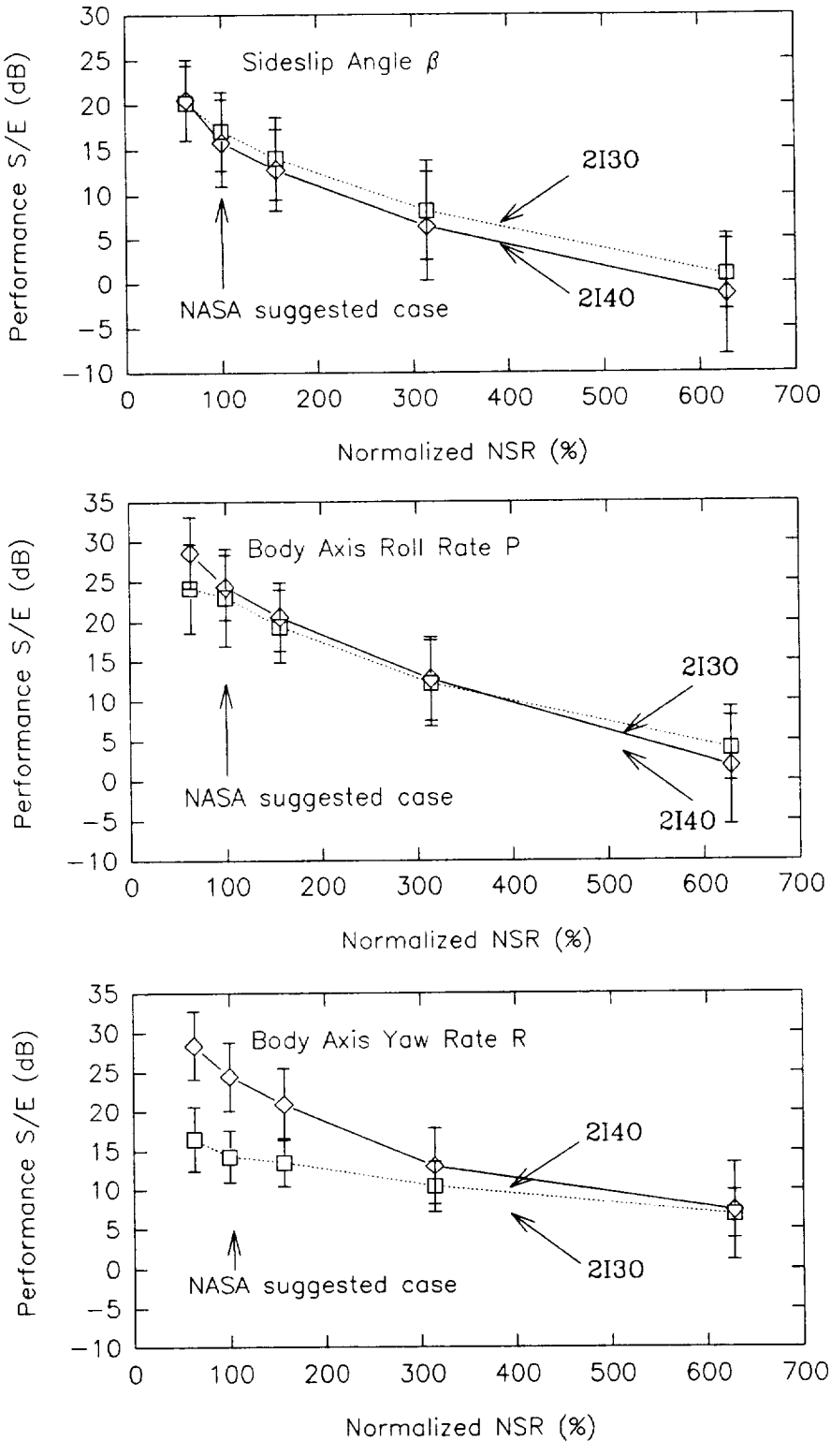

Figure 5.13: Comparison of SER numbers for 2I3O and 2I4O models

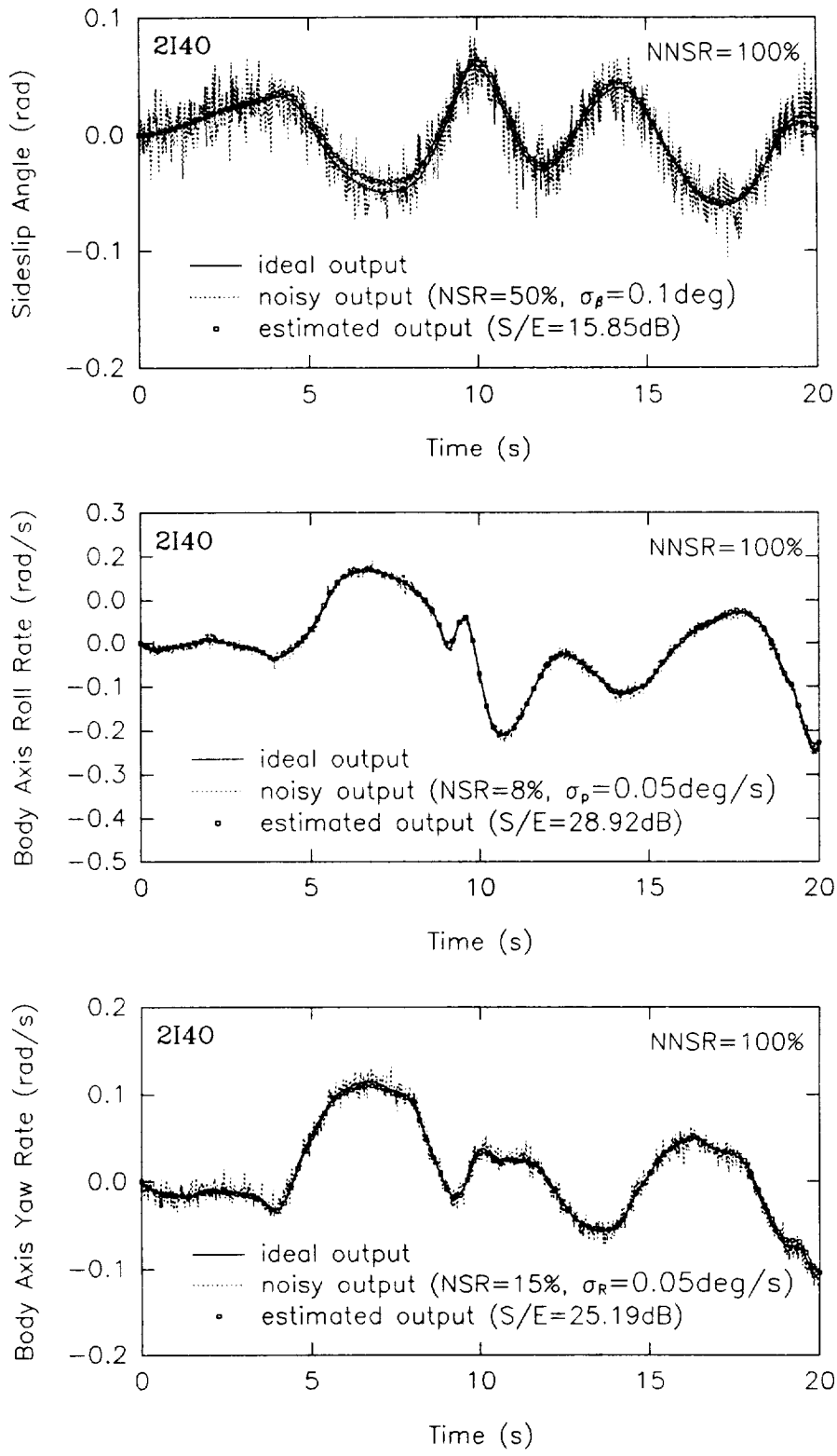

Figure 5.14: Lateral simulation at NASA suggested noise level

2I4O	denominator				Signal-to-Error Ratio (dB)				NSR_β	NSR_p	
true θ_a	0.4978	2.0426	0.5494	-0.0049	β	p	r	ϕ	NSR_r	NSR_ϕ	NNSR
mean	0.4957	2.0449	0.5458	-0.0051	20.58	28.60	28.41	26.72	32%	5%	63%
std	0.03881	0.02845	0.04582	0.00373	4.479	4.438	4.331	4.233	9%	0.6%	
mean*	0.4875	2.0450	0.5398	-0.0048	15.81	24.29	24.48	25.05	51%	8%	100%
std*	0.05198	0.04271	0.07106	0.00490	4.840	4.052	4.319	6.126	14%	1%	
mean	0.4884	2.0434	0.5414	-0.0049	12.79	20.58	20.94	19.88	80%	13%	157%
std	0.09070	0.06864	0.11525	0.00874	4.503	4.234	4.556	5.900	22%	1.6%	
mean	0.4658	2.0590	0.5233	-0.0040	6.44	12.80	13.03	12.74	160%	25%	314%
std	0.18730	0.13720	0.25562	0.01793	6.137	5.228	4.861	6.458	44%	3.1%	
mean	0.4475	2.5177	0.6517	-0.0126	-1.28	1.78	7.31	5.76	320%	50%	628%
std	0.24609	0.47250	0.51007	0.04366	6.822	7.378	6.203	7.422	88%	6.2%	

2I4O	$B_{11}(p)$				$B_{12}(p)$				
true θ_{b1}	0.0005	-1.4171	-0.3838	-0.0010	0.0098	0.3607	0.1013	-0.0096	NNSR
mean	-0.0019	-1.4074	-0.3905	0.0078	0.0087	0.3573	0.1008	-0.0132	63%
std	0.05820	0.05482	0.14572	0.04695	0.00963	0.01307	0.03070	0.02189	
mean*	-0.0045	-1.4179	-0.4042	-0.0078	0.0115	0.3598	0.1072	-0.0107	100%
std*	0.09684	0.08393	0.23247	0.08610	0.01657	0.02239	0.05456	0.03436	
mean	0.0062	-1.4258	-0.3950	-0.0160	0.0101	0.3624	0.1057	-0.0085	157%
std	0.16831	0.16294	0.35702	0.09945	0.02840	0.04245	0.07059	0.05433	
mean	-0.0734	-1.4869	-0.5868	-0.0257	0.0242	0.3630	0.1394	-0.0311	314%
std	0.31107	0.22358	0.78659	0.21128	0.05095	0.06657	0.17025	0.10618	
mean	0.0652	-1.4847	0.0506	0.1420	0.0099	0.2721	0.0562	-0.0262	628%
std	0.67416	0.64572	1.63521	0.54428	0.11964	0.21488	0.32295	0.23633	

2I4O	$B_{21}(p)$				$B_{22}(p)$				
true θ_{b2}	-3.3714	-0.1695	-0.4805	0.0151	0.3206	-0.0933	-0.9442	0.0297	NNSR
mean	-3.3631	-0.1555	-0.4740	0.0153	0.3197	-0.0936	-0.9429	0.0330	63%
std	0.06355	0.12431	0.15993	0.08470	0.01706	0.02368	0.05818	0.03427	
mean*	-3.3561	-0.1425	-0.4498	-0.0022	0.3210	-0.0956	-0.9391	0.0404	100%
std*	0.09697	0.16269	0.25232	0.11859	0.02634	0.03352	0.08786	0.04286	
mean	-3.3627	-0.1348	-0.4482	0.0176	0.3197	-0.1004	-0.9454	0.0348	157%
std	0.14111	0.29050	0.40911	0.18397	0.04014	0.05711	0.14626	0.07324	
mean	-3.3072	-0.0475	-0.4288	-0.0950	0.3184	-0.1109	-0.9042	0.0703	314%
std	0.30623	0.5539	0.90239	0.38923	0.09137	0.10302	0.31985	0.14981	
mean	-2.4245	-0.1986	0.5914	-1.7182	0.1630	0.0565	-0.8759	0.6018	628%
std	1.22100	0.96646	3.09565	1.68237	0.22311	0.31769	0.65002	0.61901	

Table 5.2: 50 Monte Carlo runs for the 2I4O system. continued to next page

2I4O	$B_{31}(p)$				$B_{32}(p)$				
true θ_{b3}	0.0512	0.0443	-0.2122	-0.0528	-0.2480	-0.1205	-0.4240	-0.1037	NNSR
mean	-0.0604	0.0467	-0.1875	-0.0496	-0.2491	-0.1190	-0.4290	-0.1004	63%
std	0.03487	0.02961	0.07364	0.02448	0.00571	0.01420	0.01677	0.01306	
mean*	-0.0530	0.0604	-0.2092	-0.0472	-0.2485	-0.1217	-0.4239	-0.1027	100%
std*	0.05347	0.03910	0.12307	0.04278	0.00804	0.01814	0.03026	0.02251	
mean	0.0510	0.0624	-0.1960	-0.0442	-0.2477	-0.1220	-0.4263	-0.1019	157%
std	0.06528	0.08418	0.17229	0.05617	0.01150	0.03605	0.04080	0.03247	
mean	0.0668	0.0445	-0.2110	-0.0420	-0.2512	-0.1116	-0.4306	-0.1010	314%
std	0.15520	0.13645	0.43128	0.12453	0.02849	0.05995	0.10341	0.06591	
mean	0.0202	0.04505	-0.2021	0.0045	-0.2403	-0.1176	-0.5743	-0.1330	628%
std	0.29986	0.25990	0.78443	0.31560	0.05071	0.09594	0.19534	0.16327	

2I4O	$B_{41}(p)$				$B_{42}(p)$				
true θ_{b4}	0.0000	-3.3567	-0.1568	-0.5413	0.0000	0.2495	-0.1278	-1.0658	NNSR
mean	0.0168	-3.3495	-0.0751	-0.5449	-0.0050	0.2492	-0.1485	-1.0566	63%
std	0.11747	0.05168	0.65505	0.05851	0.04259	0.01555	0.18734	0.09533	
mean*	0.0290	-3.3485	0.0128	-0.5366	-0.0100	0.2498	-0.1747	-1.0471	100%
std*	0.18092	0.08935	0.97541	0.07951	0.06590	0.02154	0.28038	0.14951	
mean	0.0223	-3.3494	-0.0326	-0.5339	-0.0078	0.2490	-0.1630	-1.0532	157%
std	0.29690	0.13044	1.62360	0.14707	0.10710	0.03651	0.46487	0.24128	
mean	0.0866	-3.3152	-0.3207	-0.5324	-0.0288	0.2403	-0.2543	-1.0216	314%
std	0.66787	0.33048	3.57656	0.29508	0.24097	0.07876	1.02168	0.54732	
mean	0.4973	-3.3082	1.9556	-0.8647	-0.0669	0.0767	-0.3828	-1.1494	628%
std	1.46643	0.52769	6.73534	1.52819	0.43575	0.35374	1.73141	1.30480	

NNSR is normalized NSR by the case:

$\sigma_p = 0.1$(deg){NSR=320%}, $\sigma_P = 0.05$(deg/sec) {NSR=50%},

$\sigma_R = 0.05$(deg/sec) {NSR=90%} and $\sigma_\phi = 0.01$(deg) {NSR=4%}

*means the conditions suggested by NASA.

Table 5.2: 50 Monte Carlo runs for the 2I4O system.

2I3O	denominator				Signal-to-Error Ratio (dB)				NSR_β NSR_p NSR_r		NNSR
true θ_a	0.4978	2.0426	0.5494	-0.0049	β	P	R	φ			
mean	0.5896	2.0898	0.7455	0.0365	20.24	24.16	16.50		32%	5%	63%
std	0.07444	0.03430	0.16395	0.02451	4.124	5.568	4.139		9%		
mean*	0.6384	2.1117	0.8310	0.0609	17.10	23.01	14.28		51%	8%	100%
std*	0.12285	0.08427	0.28007	0.04930	4.372	6.131	3.322		14%		
mean	0.6435	2.1299	0.8666	0.0645	14.06	19.42	13.52		80%	13%	157%
std	0.12532	0.08223	0.26781	0.04421	4.570	4.561	3.136		22%		
mean	0.7178	2.3007	1.1456	0.1165	8.23	12.25	10.42		160%	25%	314%
std	0.3840	0.22805	0.78506	0.11557	5.548	5.364	3.237		44%		
mean	0.6107	3.7172	1.6432	0.3034	0.93	4.01	6.83		320%	50%	628%
std	0.27017	0.97554	0.91253	0.23535	3.953	4.083	3.021		88%		

2I3O	$B_{11}(p)$				$B_{12}(p)$				NNSR
true θ_{b1}	0.0005	-1.4171	-0.3838	-0.0010	0.0098	0.3607	0.1013	-0.0096	
mean	-0.0040	-1.4150	-0.5423	-0.0124	0.0098	0.3594	0.1389	-0.0086	63%
std	0.06130	0.06025	0.22370	0.06063	0.01000	0.01475	0.04175	0.02495	
mean*	-0.0073	-1.4320	-0.5980	-0.0302	0.0099	0.3651	0.1529	-0.0028	100%
std*	0.11714	0.10742	0.30615	0.08245	0.02092	0.03301	0.06160	0.03911	
mean	0.0109	-1.4299	-0.5937	-0.0530	0.0119	0.3621	0.1620	0.0042	157%
std	0.17574	0.17090	0.42460	0.11126	0.03038	0.04210	0.07919	0.06003	
mean	-0.0562	-1.5255	-0.8527	0.0050	0.0248	0.3439	0.2060	0.0051	314%
std	0.31627	0.23401	1.09784	0.25488	0.05642	0.08442	0.23495	0.11614	
mean	0.0432	-0.9580	0.2054	-0.0341	0.0459	0.0591	0.3657	0.0812	628%
std	0.61253	0.52412	1.99219	0.82101	0.13793	0.16181	0.58878	0.27237	

2I3O	$B_{21}(p)$				$B_{22}(p)$				NNSR
true θ_{b2}	-3.3714	-0.1695	-0.4805	0.0151	0.3206	-0.0933	-0.9442	0.0297	
mean	-3.3101	-0.4506	-0.2708	-0.0814	0.3045	-0.0635	-1.0048	-0.0203	63%
std	0.13358	0.25478	0.58695	0.13645	0.03132	0.02744	0.14083	0.06415	
mean*	-3.3026	-0.6240	-0.3516	-0.0944	0.3061	-0.0440	-1.0032	-0.0695	100%
std*	0.17993	0.38391	0.65234	0.25284	0.04237	0.04667	0.15822	0.06924	
mean	-3.2706	-0.5983	-0.1552	-0.1597	0.2936	-0.0503	-1.0345	-0.0493	157%
std	0.24672	0.41293	0.95006	0.30031	0.05082	0.05359	0.20757	0.11916	
mean	-2.8878	-0.7190	0.5547	-0.8582	0.2331	-0.0143	-1.0607	0.0725	314%
std	0.68298	1.29488	2.20614	0.98283	0.14829	0.17885	0.50853	0.35707	
mean	-0.4345	-0.6585	3.0656	-4.6326	-0.0942	0.4349	-0.9247	1.1387	628%
std	1.88277	1.08985	5.19007	2.24115	0.33222	0.39915	1.10963	0.84094	

Table 5.3: 50 Monte Carlo runs for the 2I3O system. continued to next page

2I3O	$B_{31}(p)$				$B_{32}(p)$				
true θ_{b3}	0.0512	0.0443	-0.2122	-0.0528	-0.2480	-0.1205	-0.4240	-0.1037	NNSR
mean	0.02974	0.0407	-0.3028	-0.0124	-0.2403	-0.1358	-0.4105	-0.1658	63%
std	0.04208	0.03095	0.10482	0.04578	0.00818	0.01889	0.02032	0.04579	
mean*	0.0059	0.0414	-0.3671	0.0523	-0.2341	-0.1470	-0.3993	0.0523	100%
std*	0.06223	0.04826	0.20007	0.08745	0.01266	0.03102	0.03453	0.09147	
mean	-0.0033	0.05834	-0.3993	0.0616	-0.2321	-0.1517	-0.3957	-0.2117	157%
std	0.07577	0.08486	0.20906	0.0992	0.01485	0.03745	0.03863	0.08203	
mean	-0.0302	0.0937	-0.5789	0.1915	-0.2275	-0.1728	-0.4083	-0.3194	314%
std	0.14836	0.15328	0.45579	0.26600	0.02724	0.10844	0.09409	0.24733	
mean	-0.4184	0.3270	-1.9588	1.0482	-0.1594	-0.2433	-0.5940	1.0482	628%
std	0.41884	0.27371	1.42094	0.91631	0.07261	0.10470	0.35619	0.44713	

NNSR is normalized NSR by the case:
σ_p = 0.1(deg){NSR=51%}, σ_P = 0.05(deg/sec) {NSR=8%},
σ_R = 0.05(deg/sec) {NSR=14%} and σ_ϕ = 0.01(deg) {NSR=1%}
*means the conditions suggested by NASA.

Table 5.3: 50 Monte Carlo runs for the 2I3O system.

5.3.5 A Brief Comment on Minimal Realization

As seen in the above simulation studies, the true system is a fourth order state space model, but Algorithm 7 only returns the I/O differential models instead of state-space-looking I/O model like $(B.1)$. One question that automatically occurs to our mind is whether the estimated transfer function model, at the NASA suggested noise level through Algorithm 7, could be minimally realized using a fourth order state space model. This has been in no sense a trivial question to answer. Assuming distinct poles, the well-known Gilbert's Diagonal Realization Scheme [15] first expands the transfer function matrix $\boldsymbol{H}(s)$ into partial fractions:

$$\boldsymbol{H}(s) = \frac{\boldsymbol{N}(s)}{d(s)} = \frac{\sum_1^r \boldsymbol{N}_i s^{r-i}}{s-\lambda_i} = \sum_1^r \frac{\boldsymbol{R}_i}{s-\lambda_i} \tag{5.28}$$

where the denominator polynomial is

$$d(s) = \prod_{1}^{r}(s - \lambda_i)$$

and $\{\boldsymbol{R}_i\}$ are the residue matrices. Further denote

$$\rho_i = \text{the rank of } \boldsymbol{R}_i$$

and then Gilbert's method says that the minimal realization has order

$$n = \sum_{1}^{r} \rho_i \tag{5.29}$$

So the question of obtaining a minimal realization of the transfer function model $\boldsymbol{H}(s)$ has boiled down to the determination of the residue matrices $\{\boldsymbol{R}_i\}$ and their ranks. As suggested by Professor A.E. Pearson, employing the partial fraction expansion algorithm in [19] and the rank determination scheme based on the singular value decomposition technique [10], some verifications on $\{\boldsymbol{R}_i\}$ and $\{\rho_i\}$ have been computed. Define the tolerance for rank determination[8] by:

$$\delta_{SVD} = \bar{\sigma}_{max} \cdot \|\boldsymbol{R}_i\|_\infty$$

where $\bar{\sigma}_{max}$ is the precision of estimating the matrix elements, i.e., the maximum standard deviation value among all the entries comprising each $\boldsymbol{N}_i$ matrix. This is obtained from the std values in Table 5.2$\sim$ 5.2. At the NASA suggested noise level, our calculations on the models estimated by Algorithm 7 with different additive noise realizations show that all the $\{\boldsymbol{R}_i\}$ have persistently been diagnosed as rank one

[8]All the singular values less than δ_{SVD} will be dropped.

matrices. This means that the estimated transfer function model can be realized with a fourth order state space model and demonstrates consistency with the theoretical state space model.

5.4 Miscellany

Inasmuch as many other issues pertaining to the F-18 dynamics have been investigated as well, some relatively-peripheral-but-intriguing results should be worth mentioning briefly in this section. Hopefully, some merit of the SISO AWLS/MFT Algorithm 2 could be further divulged through handling the physical flight data.

5.4.1 Identification of the Actuator System

The F-18 horizontal tail deflection $\delta_h(t)$ is activated by the longitudinal pilot stick movement $\eta_h(t)$ through a mechanical actuator system. Using the SISO AWLS/MFT Algorithm 2 in Section 2.2.3, the dynamics of this actuator system can be modeled as a linear second order system resulting in the transfer function:

$$\frac{\delta_h(s)}{\eta_h(s)} = \frac{-0.0494s^2 - 0.0462s - 0.0340}{s^2 + 0.2586s + 1.2987}. \tag{5.30}$$

The time domain performance of the above model has been plotted in Figure 5.15 from which we can see that the model yields an impressive time domain fit. As to why a second order model is used, several other order model structures were also tested and tabulated. The model (5.30) results from using the parsimony principle:

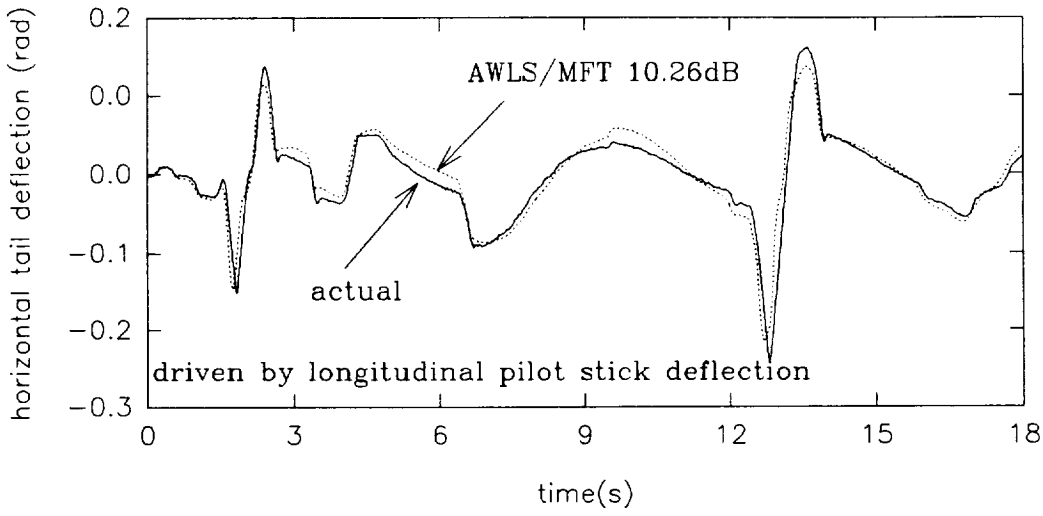

Figure 5.15: Time domain performance of the estimated longitudinal actuator

pick the lowest order model among all the models which give a reasonable time domain performance. Together with the time domain performance, other approaches like statistically checking the whiteness of the residual process $W^{-1/2}(Y - \Gamma\hat{\theta})$ using Lemma 1 with a 95% confidence bound [5] [52] had also been adopted to determine the model structures in our early research report to NASA.

5.4.2 Sensitivity to the Chosen Modulating Bandwidth F_B

As mentioned earlier, our experience is that the AWLS/MFT Algorithm is far less sensitive to the chosen modulating bandwidth F_B than either the LS/MFT or WLS/MFT algorithm. Pearson and Lee in [39] found that overmodulating (broader modulating bandwidth than the system bandwidth) could exaggerate the estimation errors when LS/MFT is used. In our initial studies on the longitudinal dynamics, a fourth order SISO model was once built to link the longitudinal pilot stick movement to the

body axis pitch rate $q(t)$. The physical spectrum of the body pitch rate $q(t)$ has a bandwidth around 0.5Hz. During this model-building process, a set of modulating bandwidths ranging from 0.3Hz~1.0Hz had been chosen for LS/MFT, WLS/MFT using both the unconstrained and constrained AWLS/MFT algorithms. The results are summarized in Table 5.4. It is seen that the models for the LS/MFT and WLS/MFT

Sensitivity of different algorithms to the modulating bandwidth F_B

algorithm $F_B(Hz)/M$	AWLS constrained	AWLS unconstrained	WLS	LS
0.3/12	**stable** s/e=5.84dB	**stable** s/e=5.84dB	unstable s/e= -4.78dB	**stable** s/e= -6.79dB
0.4/16	**stable** s/e=6.83dB	**stable** s/e=6.83dB	unstable s/e= 2.09dB	unstable s/e= -3.39dB
0.5/20	**stable** s/e=7.13dB	**stable** s/e=7.13dB	**stable** s/e=1.86dB	unstable s/e=1.53dB
0.6/24	**stable** s/e=6.91dB	unstable s/e=7.56dB	**stable** s/e=3.08dB	unstable s/e= -7.41dB
0.7/28	**stable** s/e=6.51dB	unstable s/e=8.07dB	**stable** s/e=3.32dB	unstable s/e= -53.22dB
0.8/32	**stable** s/e=6.41dB	unstable s/e=8.10dB	unstable s/e= -27.63dB	unstable s/e= -41.04dB
0.9/36	**stable** s/e=6.56dB	unstable s/e=7.94dB	unstable s/e= -33.98dB	unstable s/e= -27.60dB
1.0/40	**stable** s/e=6.47dB	unstable s/e=8.00dB	unstable s/e= -61.97dB	unstable s/e= -121.1dB

Using physical flight data to build a fourth order model linking the longitudinal pilot stick movement (input) and the body pitch rate (output).

Table 5.4: Sensitivity of MFT algorithms to the chosen modulating bandwidth

are either stable with poor SER numbers or unstable with unacceptable SER's. On the other hand, the estimated systems from both the constrained and unconstrained AWLS/MFT algorithms yield much more consistent time domain performance with

no drastic fluctuations like the LS/MFT and WLS/MFT, no matter whether the estimated models are stable or not. All these help claim that the AWLS/MFT algorithm not only performs better, but also has less sensitivity to the pre-chosen modulating bandwidth F_B.

Appendix A

Longitudinal Model from NASA

For simulation purpose, the following linearized theoretical longitudinal models coupling $\alpha(t)$ and $q(t)$ are given by NASA:

$$\begin{cases} \dot{\alpha}(t) &= -0.1692\alpha(t) + 0.9560q(t) - 0.0520\delta_h(t) \\ \dot{q}(t) &= -0.3141\alpha(t) - 0.3403q(t) - 1.5358\delta_h(t) \end{cases} . \tag{A.1}$$

The corresponding constrained differential dynamical model is

$$\begin{cases} A(p)\alpha(t) = B_1(p)\delta_h(t) \\ A(p)q(t) = B_2(p)\delta_h(t) \end{cases} \tag{A.2}$$

where

$$A(p) = \begin{pmatrix} 1.0000 & 0.5095 & 0.3579 \end{pmatrix} \begin{pmatrix} p^2 \\ p \\ 1 \end{pmatrix} \tag{A.3}$$

$$\begin{pmatrix} B_1(p) \\ B_2(p) \end{pmatrix} = \begin{pmatrix} -0.0520 & -1.4860 \\ -1.5358 & -0.2435 \end{pmatrix} \begin{pmatrix} p \\ 1 \end{pmatrix} . \tag{A.4}$$

Appendix B

Lateral Model from NASA

For simulation purpose, the following linearized theoretical lateral dynamical models coupling $\beta(t)$, $p(t)$, $r(t)$ and $\phi(t)$ are given by NASA:

$$\begin{pmatrix} \dot{\beta}(t) \\ \dot{p}(t) \\ \dot{r}(t) \\ \dot{\phi}(t) \end{pmatrix} = \begin{pmatrix} -0.0432 & 0.4066 & -0.9108 & 0.1003 \\ -4.4588 & -0.4441 & 0.2692 & 0 \\ 0.2239 & -0.0057 & -0.0106 & 0 \\ 0 & 1.000 & 0.2867 & 0 \end{pmatrix} \begin{pmatrix} \beta(t) \\ p(t) \\ r(t) \\ \phi(t) \end{pmatrix} + \begin{pmatrix} 0.0005 & 0.0098 \\ -3.3714 & 0.3206 \\ 0.0512 & -0.2480 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \delta_a(t) \\ \delta_r(t) \end{pmatrix}. \tag{B.1}$$

The corresponding constrained differential dynamical model is

$$\begin{cases} A(p)\beta(t) = B_{11}(p)\delta_a(t) + B_{12}(p)\delta_r(t) \\ A(p)p(t) = B_{21}(p)\delta_a(t) + B_{22}(p)\delta_r(t) \\ A(p)r(t) = B_{31}(p)\delta_a(t) + B_{32}(p)\delta_r(t) \\ A(p)\phi(t) = B_{41}(p)\delta_a(t) + B_{42}(p)\delta_r(t) \end{cases} \tag{B.2}$$

where

$$A(p) = \begin{pmatrix} 1.0000 & 0.4978 & 2.0426 & 0.5494 & -0.0049 \end{pmatrix} \begin{pmatrix} p^4 \\ p^3 \\ p^2 \\ p \\ 1 \end{pmatrix} \tag{B.3}$$

$$\begin{pmatrix} B_{11}(p) \\ B_{21}(p) \\ B_{31}(p) \\ B_{41}(p) \end{pmatrix} = \begin{pmatrix} 0.0005 & -1.4171 & -0.3838 & -0.0010 \\ -3.3714 & -0.1695 & -0.4805 & 0.0151 \\ 0.0512 & 0.0443 & -0.2122 & -0.0528 \\ 0.0000 & -3.3567 & -0.1568 & -0.5413 \end{pmatrix} \begin{pmatrix} p^3 \\ p^2 \\ p \\ 1 \end{pmatrix} \tag{B.4}$$

$$\begin{pmatrix} B_{12}(p) \\ B_{22}(p) \\ B_{32}(p) \\ B_{42}(p) \end{pmatrix} = \begin{pmatrix} 0.0098 & 0.3607 & 0.1013 & -0.0096 \\ 0.3206 & -0.0933 & -0.9442 & 0.0297 \\ -0.2480 & -0.1205 & -0.4240 & -0.1037 \\ 0.0000 & 0.2495 & -0.1278 & -1.0658 \end{pmatrix} \begin{pmatrix} p^3 \\ p^2 \\ p \\ 1 \end{pmatrix}. \tag{B.5}$$

Appendix C

Initial Condition (I.C.) Estimate via Luenberger Observor

The following algorithm for estimating I.C. is suggested by Professor A.E. Pearson. Given $(\hat{A}(p), \hat{B}(p))$ for the differential model $\hat{A}(p)y(t) = \hat{B}(p)u(t), 0 \leq t \leq T$, with the I/O data pair $[u(t), y(t)]$ on $[0, T]$, we want to find $x_0 = x(0)$ for the state realization

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \qquad \text{and} \qquad x_0 = x(0) \end{aligned}$$

where (A, B, C) is any observable state space realization for $\frac{\hat{B}(s)}{\hat{A}(s)}$, i.e.,

$$C(sI - A)^{-1}B = \frac{\hat{B}(s)}{\hat{A}(s)}.$$

The I.C. estimate can be constructed through the following steps:

1. Subtract off the zero-state response $y_u(t) = \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau$ from $y(t)$:

$$y(t) - y_u(t) = Ce^{At}x_0 \qquad \text{I.C. response}$$

2. Define the reversed I.C. response

$$\begin{aligned} z(t) &= y(T-t) - y_u(T-t) \\ &= Ce^{-At}e^{AT}x_0 \\ &= Cx_R(t) \end{aligned} \tag{C.1}$$

where $x_R(t) = e^{-At}e^{AT}x_0$ satisfies: $\dot{x}_R(t) = -Ax_R(t)$, $x_R(0) = e^{AT}x_0$, and more importantly $x_R(T) = x_0$. Hence, we estimate $x_R(t)$ given $z(t)$ via an observer as follows:

$$\begin{aligned} \dot{\hat{x}}_R(t) &= -A\hat{x}_R(t) + L(z(t) - C\hat{x}_R(t)) \\ &= -(A + LC)\hat{x}_R(t) + Lz(t) \end{aligned}$$

with $\hat{x}_R(0)$ an arbitrary estimate. In order to carry out the error analysis, let $\tilde{x}(t) = x_R(t) - \hat{x}_R(t)$, then

$$\begin{aligned} \dot{\tilde{x}}(t) &= -Ax_R(t) - [-A\hat{x}_R(t) + L(Cx_R(t) - C\hat{x}_R(t))] \\ &= -A(x_R(t) - \hat{x}_R(t)) - LC(x_R(t) - \hat{x}_R(t)) \\ &= -(A + LC)\tilde{x}(t) \end{aligned}$$

which implies

$$\tilde{x}(t) = e^{-(A+LC)t}\tilde{x}(0) \longrightarrow 0$$

provided $-(A + LC)$ is Hurwitzian.

3. Design the gain matrix L such that $e^{-(A+LC)T} \approx 0$ and use $\hat{x}_R(T)$ as an estimate of x_0, i.e., $\hat{x}_R(T) \approx x_0$.

Appendix D

Some Results in M.S. Differentiation and Integration

The following results are cited mainly from the Chapter 4 of [53].

D.1 Mean Square Continuity

Definition 1 (continuous in mean square) *A second-order stationary (or wide sense stationary) s.p. $X(t)$, $t \in T$, is continuous in mean square, or m.s. continuous, at a fixed t if*

$$l.i.m._{\tau \to 0} X(t+\tau) = X(t) \tag{D.1}$$

Theorem 1 (continuity in mean square criterion) *A second-order s.p. $X(t)$, $t \in T$, is m.s. continuous at t if, and only if, $\Gamma(t,s) = E\{X(t)X(s)\}$ is continuous at (t,t).*

Definition 2 (m.s. continuous on an interval) *If a second-order s.p.* $X(t)$, $t \in T$, *is m.s. continuous at every* $t \in [t_1, t_2] \subset T$, *then* $X(t)$ *is m.s. continuous on the interval* $[t_1, t_2]$.

D.2 Mean Square Differentiation

Definition 3 (m.s. derivative) *A second-order s.p.* $X(t)$, $t \in T$, *has a mean square derivative (or m.s. derivative)* $\dot{X}(t)$ *at* t *if*

$$l.i.m._{\tau \to 0} \frac{X(t+\tau) - X(t)}{\tau} = \dot{X}(t) \tag{D.2}$$

Definition 4 (differentiable on an interval) *If a second-order s.p.* $X(t)$, $t \in T$, *is m.s. differentiable at every* $t \in [t_1, t_2] \subset T$, *then* $X(t)$ *is m.s. differentiable on the interval* $[t_1, t_2]$.

Theorem 2 (criterion of m.s. differentiable) *A wide sense stationary second-order process* $X(t)$, $t \in T$, *is m.s. differentiable if, and only if, the first- and second-order derivatives of the correlation function* $\Gamma(t,s) = E\{X(t)X(s)\}$ *exist and are finite at* $t - s = 0$.

D.3 Properties of Mean Square Derivatives

1. Mean square differentiability of $X(t)$ at $t \in T$ implies m.s. continuity of $X(t)$ at t.

2. The m.s. derivative $\dot{X}(t)$ of $X(t)$ at $t \in T$, if it exists, is unique.

3. If $X(t)$ and $Y(t)$ are m.s. differentiable at $t \in T$, then the m.s. derivative of $aX(t) + bY(t)$ exists at t and

$$\frac{d}{dt}[aX(t) + bY(t)] = a\dot{X}(t) + b\dot{Y}(t) \tag{D.3}$$

where a and b are constants.

4. If an ordinary function $f(t)$ is differentiable at $t \in T$ and $X(t)$ is m.s. differentiable at $t \in T$, then $f(t)X(t)$ is m.s. differentiable at t and

$$\frac{d}{dt}[f(t)X(t)] = \frac{df(t)}{dt}X(t) + f(t)\frac{dX(t)}{dt} \tag{D.4}$$

D.4 Mean Square Riemann Integration

Consider a collection of all finite partitions $\{p_n\}$ of an interval $[a, b]$. The partition p_n is defined by the subdivision points t_k, $k = 0, 1, 2, ..., n$, such that

$$a = t_0 < t_1 < t_2 < \cdots < t_n = b$$

Let

$$\Delta_n = \max_k(t_k - t_{k-1})$$

and let t'_k be an arbitrary point in the interval $[t_{k-1}, t_k)$. Let $X(t)$ be a second-order s.p. defined on $[a, b] \subset T$. Let $f(t, u)$ be an ordinary function defined on the same interval for $t \in [a, b]$ and Riemann integrable for every $u \in U$. We form the random

variable

$$Y_n(u) = \sum_{k=1}^{n} f(t'_k, u)X(t'_k)(t_k - t_{k-1})$$

Since L_2-space is linear, $Y_n(u)$ is an element of the L_2-space. It is a random variable defined for each partition p_n and for each $u \in U$.

Definition 5 (m.s. Riemann integral) *If, for* $u \in U$,

$$l.i.m._{n\to\infty \Delta_n \to 0} Y_n(u) = Y(u)$$

exists for some sequence of subdivisions p_n, *the s.p.* $Y(u)$, $u \in U$, *is called the mean square Riemann integral, or m.s. Riemann integral of* $f(t,u)X(t)$, *over the interval* $[a,b]$, *and it is denoted by*

$$Y(u) = \int_a^b f(t,u)X(t)dt \tag{D.5}$$

It is independent of the sequence of subdivisions as well as the positions of the $t'_k \in [t_{k-1}, t_k)$.

Theorem 3 (integration in mean square criterion) *The s.p.* $Y(u)$, $u \in U$, *defined by Eq.* (D.5) *exists if, and only if, the ordinary double Riemann integral*

$$\int_a^b \int_a^b f(t,u)f(s,u)\Gamma_{XX}(t,s)dtds$$

exists and is finite.

D.4.1 Properties of mean square Riemann integrals

1. Mean square continuity of $X(t)$ on $[a,b]$ implies m.s. Riemann integrability of $X(t)$ on $[a,b]$.

2. The m.s. integral of $X(t)$ on an interval $[a, b]$, if it exists, is unique.

3. If $X(t)$ is m.s. continuous on $[a, b]$, then

$$\left\| \int_a^b X(t)dt \right\| \leq \int_a^b \|X(t)\|dt \leq M(b-a) \tag{D.6}$$

where

$$M = \max_{t \in [a,b]} \|X(t)\|$$

4. If the m.s. integrals of $X(t)$ and $Y(t)$ exist on $[a, c]$, then

$$\int_a^c [\alpha X(t) + \beta Y(t)]dt = \alpha \int_a^c X(t)dt + \beta \int_a^c Y(t)dt \tag{D.7}$$

$$\int_a^c X(t)dt = \int_a^b X(t)dt + \int_a^c X(t)dt, \qquad a \leq b \leq c \tag{D.8}$$

5. If $X(t)$ is m.s. continuous on $[a, t] \subset T$, then

$$Y(t) = \int_a^t X(t)dt$$

is m.s. continuous on T; it is also m.s. differentiable on T with

$$\dot{Y}(t) = X(t)$$

Corollary 1 (Leibniz Rule) *If $X(t)$ is m.s. integrable on T and if the original function $f(t, s)$ is continuous on $T \times T$ with a finite first partial derivative $\partial f(t, s)/\partial t$, then the m.s. derivative of*

$$Y(t) = \int_a^t f(t, s)X(s)ds$$

exists at all $t \in T$, and

$$\dot{Y}(t) = \int_a^t \frac{\partial f(t, s)}{\partial t} X(s)ds + f(t, t)X(t) \tag{D.9}$$

Corollary 2 (Integration by Parts) *Let $X(t)$ be m.s. differentiable on T and let the ordinary function $f(t,s)$ be continuous on $T \times T$ whose partial derivative $\partial f(t,s)/\partial t$ exists. If*

$$Y(t) = \int_a^t f(t,s)\dot{X}(s)ds \tag{D.10}$$

then

$$Y(t) = f(t,s)X(s)|_a^t - \int_a^t \frac{\partial f(t,s)}{\partial s} X(s)ds \tag{D.11}$$

Let $f(t,s) \equiv 1$ in Eqs. $(D.10)$ and $(D.11)$; we have

$$X(t) - X(a) = \int_a^t \dot{X}(s)ds, \qquad [a,t] \in T$$

This property is seen to be the m.s. counterpart of the fundamental theorem of the ordinary calculus.

D.4.2 Means and Correlation Functions of M.S. Riemann Integrals

Corollary 3 (means and correlations of m.s. Riemann Intetral) *If the m.s. Riemann integral*

$$Y(u) = \int_a^b f(t,u)X(t)dt$$

exist, then

$$E\{Y(u)\} = \int_a^b f(t,u)E\{X(t)\}dt \tag{D.12}$$

and

$$E\{Y(u)Y(v)\} = \int_a^b \int_a^b f(t,u)f(s,v)E\{X(t)X(s)\}dtds \tag{D.13}$$

Bibliography

[1] K.J. Astrőm. Maximum likelihood and prediction methods. *Trends and Progress in System Identification, ed. by P. Eykhoff, Pergamon Press, Oxford*, pages 145–168, 1981.

[2] K.J. Astrőm and B. Wittenmark. *Adaptive Control.* Addison-Wesley Publishing Company, 1989.

[3] R.E. Bellman and R. Kalaba. *Quasilinearization and Nonlinear Boundary Value Problems.* American Elsevier, New York, 1965.

[4] J.H. Blakelock. *Automatic Control of Aircraft and Missiles.* John Wiley & Sons, Inc. 2nd ed., 1991.

[5] G.E.P. Box and G.M. Jenkins. *Time Series Analysis-Forecasting and Control.* Holden Day, San Francisico, 1970.

[6] T.B. Co and B.E. Ydstie. System identification using modulating functions and fast fourier transforms. *Computers Chem. Engng*, 14, No.10:1051–1066, 1990.

[7] F.W. Fairman and D.W.C.Shen. Parameter identification for linear time varying dynamic processes. *Proc. Inst. Elect. Eng.*, 117:2025–2029, 1970.

[8] A. Fullerton Jr. private communication. 1990.

[9] G.H. Golub and C.F. Van Loan. *Matrix Computatuions.* the Johns Hopkins University Press., 1983.

[10] G.H. Golub and C.F. Van Loan. *Matrix Computatuions.* the Johns Hopkins University Press., 1989.

[11] G.C. Goodwin and R.L. Payne. *Dynamic System Identification:experiment design and data analysis.* Academic Press, 1977.

[12] G.C. Goodwin and K.S. Sin. *Adaptive Filtering, Prediction and Control.* Prentice Hall, 1984.

[13] I.S. Gradshteyn and I.M. Ryzhik. *Table of Integrals, Series, and Products.* Academic Press, 1980.

[14] X. Hu. Ff-padè method of model reduction in frequency domain. *IEEE Tranc. Automatic Control*, AC-32:243–246, 1987.

[15] T. Kailath. *Linear Systems.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1980.

[16] R. Kalaba and K. Springarn. *Control, Identification and Input Optimization.* Plenum, New York, 1982.

[17] H.J. Larson and B.O.Shubert. *Probabilistic Models in Engineering Sciences.*, volume 2, Random Noise, Signals, and Dynamic Systems. John Wiley and Sons, Inc, 1979.

[18] F.C. Lee. *Time limited identification of continuous systems using the modulating functions method.* Ph.D. thesis, Division of Engineering, Brown University, Providence, Rhode Island, 1984.

[19] J. Leyva-Ramos. Partial-fraction expansion in system analysis. *Int. J. Control*, 53(3):619–639, 1991.

[20] L. Ljung. *System Identification: Theory for the User.* Prentice Hall International Ltd., 1987.

[21] L. Ljung. *System Identification Toolbox, Version 2.11.* The Math Works, Inc., 1991.

[22] L. Ljung and T. Söderström. *Theory and Practice of Recursive Identification.* MIT Press, 1983.

[23] T.N. Lucas. Some further observations on the differentiation method of model reduction. *IEEE Tranc. Automatic Control*, AC-37:1389–1391, 1992.

[24] B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Tranc. Automatic Control*, AC-26:17–32, 1981.

[25] R. Nagarajan. Optimal reduction of large dynamic systems. *Int. J. Contr.*, vol-14:1169–1174, 1971.

[26] J.Q. Pan. *System identification, model reduction and deconvolution filtering using Fourier based modulating signals and high order statistics.* Ph.D. thesis, Division of Engineering, Brown University, Providence, Rhode Island, 1992.

[27] J.Q. Pan and A.E. Pearson. High resolution frequency estimation in the presence of noise using complex sinusoidal modulating signals. *submitted to IEEE Trans. on Signal Processing.*

[28] A.E. Pearson. Finite time interval linear system identification without initial state estimation. *Automatica*, 12:577–587, 1976.

[29] A.E. Pearson. Nonlinear system identification with limited time data. *Automatica*, 15:73–84, 1979.

[30] A.E. Pearson. Least squares parameter identification of nonlinear differential i/o models. *Proc. of 27th IEEE Conf. on Dec. and Contr., Austin, TX*, pages 1831–1835, 1988.

[31] A.E. Pearson. On the equation error identification of nonlinear differential systems. *Proc. of the 1988 Conference on Information Science and System, Princeton University, Princeton, N.J.*, 1988.

[32] A.E. Pearson. Identifiability and well-posedness in nonlinear system i/o modeling. *Proc. of 27th IEEE Conf. on Dec. and Contr., Tampa, Florida*, 1989.

[33] A.E. Pearson. Explicit parameter identification for a class of nonlinear input/output differential operator models. *IEEE CDC, invited*, 1992.

[34] A.E. Pearson. Explicit least square system identification for exact differential input/output models. *Proc. of 8th ICMCM, X.J.R. Avula, ed., Pergamon Press, Oxford, UK*, to appear.

[35] A.E. Pearson and F.C.Lee. Efficient parameter identification for a class of bilinear differential systems. *Proc. of 7th IFAC/IFORS Symp. on ident. and Syst. Param. Est., York, United Kingdom*, pages 161–165, 1985.

[36] A.E. Pearson and J.Q.Pan. Frequency analysis via the method of moment functionals. *Proc. of the 30th IEEE CDC, Brighton, England*, Dec. 1991.

[37] A.E. Pearson and F.C. Lee. Time limited identification of continuous systems using trigonometric modulating functions. *Proc. of Third Yale Workshop on Applications of Adaptive Systems, New Haven, Connecticut*, pages 168–173, 1983.

[38] A.E. Pearson and F.C. Lee. On the identification of polynomial input-output differential systems. *IEEE Trans. Automatic Control*, AC-30:778–782, 1985.

[39] A.E. Pearson and F.C. Lee. Parameter identification of linear differential systems via the fourier based modulating functions. *Control-Theory and Adv. Tech.*, 1, No.4:239–266, 1985.

[40] A.E. Pearson, Y. Shen, and J.Q. Pan. Discret frequency formats for linear differential system identification. *accepted by 1993 IFAC in press.*

[41] Jon Pumplin. Low-noise noise. *J. Acoust. Soc. Am.*, 78(1), July, 1985.

[42] P.A. Ruymgaart and T.T. Soong. *Mathematics of Kalman-Bucy Filtering.* Springer-Verlag, 1985.

[43] D.C. Saha and G.P. Rao. Time domain synthesis via poisson moment functionals. *Int. J. Control*, 30:417–420, 1979.

[44] D.C. Saha and G.P. Rao. Identification of lumped linear systems in the presence of unknown initial conditions via poisson moment functional approach. *Int. J. Control*, 31:637–644, 1980.

[45] D.C. Saha and G.P. Rao. General algorithm for parameter identification in lumped continuous systems - the poisson moment functional approach. *IEEE Tranc. Automatic Control*, AC-27:223–225, 1982.

[46] D.C. Saha and G.P. Rao. *Identification of Continuous Dynamical Systems. Lecture Notes in Control and Information Science.* Springer-Verlag, Berlin, 1983.

[47] R. Sakr and A. Bahgat. Optimal reduced order model for two generating stations. *IFAC Large Scale Systems: theory and applications*, vol-1:123–127, 1986.

[48] Y. Shamash. Multivariable system reduction via model methods and padè approximation. *IEEE Tranc. Automatic Control*, AC-20:815–817, 1975.

[49] Y. Shen and A.E. Pearson. Recursive parametric and nonparametric system identification using modulating function technique. *Tech. report of Laboratory for Engineering Man/Machine Systems (LEMS), Division of Engineering, Brown University*, in final preparation.

[50] L.S. Shieh and Y.J. Wei. A mixed method for multivariable system reduction. *IEEE Tranc. Automatic Control*, AC-20:429–432, 1975.

[51] M. Shinbrot. On the analysis of linear and nonlinear system. *Trans. ASME*, 79:547–552, 1957.

[52] N.K. Sihna and B. Kuszta. *Modeling and Identification of Dynamic Systems.* Van Nostrand Reihold Company, 1983.

[53] T.T. Soong. *Random Differential Equations in Science and Engineering.* Academic Press, 1973.

[54] Sőderstrőm T. and P. Stocia. *System Identification.* Prentice Hall International Ltd., 1989.

[55] K Takaya. The use of hermite functions for system identification. *IEEE Trans. Automatic Control*, AC-13:446–447, 1968.

[56] P.C. Young. Parameter estimation for continuous-time models - a survey. *Automatica*, 17:23–39, 1981.

[57] V. Zakin. Simplication of linear time-invariant system by moment approximation. *Int. J. Contr.*, vol-18:455–460, 1973.